Multiple Model Adaptive Estimator Target Tracker For Maneuvering Targets in Clutter

THESIS

Brian D. Smith, Captain, USAF

AFIT/GE/ENG/05-18

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/05-18

Multiple Model Adaptive Estimator Target Tracker For Maneuvering Targets in Clutter

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Brian D. Smith, B.S.E.E.

Captain, USAF

March 2005

AFIT/GE/ENG/05-18

Multiple Model Adaptive Estimator Target Tracker For Maneuvering Targets in Clutter

Brian D. Smith, B.S.E.E.

Captain, USAF

Approved:

/signed/	11 Mar 2005
Dr Peter S. Maybeck (Chairman)	date
/signed/	11 Mar 2005
Lt Col Juan Vasquez (Member)	date
/signed/	11 Mar 2005
Dr Meir Pachter (Member)	date

Abstract

The task of tracking a target in the presence of measurement clutter is a two-fold problem: one of handling measurement association uncertainty (due to clutter) and poorly-known or significantly varying target dynamics. Measurement association uncertainty does not allow conventional tracking algorithms (such as Kalman filters) to be implemented directly. Poorly-known or varying target dynamics complicate the design of any tracking filter, and filters using only a single dynamics model can rarely handle anything beyond the most benign target maneuvers.

In recent years, the Multiple Hypothesis Tracker has gained acceptance as means of handling targets in a measurement-clutter environment. MHT algorithms rely on Gaussian mixture representations of a target's current state estimate, and the number of components within these mixtures grows exponentially with each successive sensor scan. Previous research into techniques that limit the growth of Gaussian mixture components proved that the Integral Square Error cost-function-based algorithm performs well in this role. Also, multiple-model adaptive algorithms have been shown to handle poorly-known target dynamics or targets that exhibit a large range of maneuverability over time with excellent results.

This research integrates the ISE mixture reduction algorithm into Multiple-Model Adaptive Estimator (MMAE) and Interacting Mixed Model (IMM) tracking algorithms. The algorithms were validated to perform well at a variety of measurement clutter densities by using a Monte Carlo simulation environment based on the C++ language. Compared to single-dynamics-model MHT trackers running against a maneuvering target, the Williams-filter-based multiple-model algorithms exhibited superior tracking performance.

Acknowledgements

First and foremost, I want to remember the words of Jesus Christ, who said, "If a man remains in me and I in him, he will bear much fruit; apart from me you can do nothing." *(John 15:5, NIV)*

I owe a tremendous debt of gratitude to Dr. Peter Maybeck, my thesis advisor, for his guidance, encouragement, wisdom, and expert knowledge. His contribution to my success at AFIT cannot be overstated. I want to thank my wife, who patiently tolerated many late night homework sessions, my commandeering of the home computer, and all of the other aggravations that are part of being an AFIT spouse. She is the most important person in my life, and I look forward to reminding her of this now that my schooling is complete. I also look forward to more playtime with my baby girl, who right now must believe her daddy only reads math books.

I owe specific thanks to my fellow classmates in the Guidance, Navigation, and Control curriculum. Lt. Hanson, who became one of my best friends, offered excellent conversation, a clever sense of humor, and a sympathetic ear at just the right times. Mike Starr, our unsuspecting Test Pilot School student, had a positive attitude that I wish I could claim as my own. Maj Meidel, a man of God who exhibited skill in the classroom and wisdom in life, was both an encouragement and role model for all of us.

Finally, I thank the faculty and staff at AFIT, especially the professors in the GNC group, who make this a truly world-class institution. I know this time at AFIT will be a highlight in my Air Force career.

Brian D. Smith

Table of Contents

List of Figures

Figure Page

Notation

Notation	Usage
$\boldsymbol{z}$, $\boldsymbol{Z}$, etc.	vectors are shown in boldface italic text
$\mathbf{P}$, $\mathbf{H}$, etc.	matrices are shown in boldface roman text
$\hat{\boldsymbol{x}}$, etc.	estimates are indicated using the 'hat' augmentation
$\boldsymbol{x}(t)$	a continuous-time signal, where the indexing t is a continuous variable representing the time in seconds
$\boldsymbol{x}(k)$	a discrete-time signal, where the indexing k is the sample number, and the k-th sample is taken at time t_k
$\hat{\boldsymbol{x}}(k\|k-1)$	the estimate of the signal at sample k, using information only up to the $(k-1)$-th measurement
$\hat{\boldsymbol{z}}(k\|k-1)$	the predicted value of the measurement at sample k, using information only up to the $(k-1)$-th measurement
$\boldsymbol{Z}^k$	the entire measurement history from sample 1 to sample k
$\boldsymbol{Z}_k$	*all* measurements provided to the system in the k-th set of measurements (i.e., the k-th scan)
$\boldsymbol{z}_j(k)$	the j-th measurement from the k-th set of measurement (i.e., the k-th scan)
$\boldsymbol{\zeta}_k$	the particular measurement on the k-th sample period
$\boldsymbol{\mathcal{Z}}^k$	a particular measurement history from sample 1 to sample k
$\boldsymbol{a}$	a vector of uncertain parameters in a dynamic system
$\boldsymbol{a}_m$	the m-th vector of discretized parameters in a dynamic system containing multiple models representing a discretized parameter space
$\boldsymbol{\alpha}$	the particular set of parameter values within a discretized parameter vector $\boldsymbol{a}_m$
$P\{\cdot\}$	the probability of the discrete event specified in $\{\cdot\}$
$f\{\cdot\}$	the probability density function of the continuous parameter specified in the argument $\{\cdot\}$
$\mathcal{N}\{\boldsymbol{x};\boldsymbol{\mu},\mathbf{P}\}$	denotes a Gaussian probability density function for variable $\boldsymbol{x}$, distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$: $\mathcal{N}\{\boldsymbol{x};\boldsymbol{\mu},\mathbf{P}\} = \lvert 2\pi\mathbf{P}\rvert^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\mathbf{P}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$
$E\{\cdot\}$	the expectation operation, finding the expected value of the argument: $E\{\cdot\} = \int \{\cdot\} f\{\cdot\}\, \mathrm{d}\{\cdot\}$
N_f	the number of filters (or models) in the system
M_j	the event in which model j is in force in a non-switching multiple model system (no time argument is supplied, as this is the non-switching case, in which the model in force does not change with time)
$M_{k,j}$	the event in which model j is in force at sample k in a switching multiple model system
$M^{k,l}$	the l-th model history event in a switching multiple model system — consists of a single time step event (e.g., $M_{k,j}$) for each sample time from 1 to k
$\mathbf{T}(k\|k-1)$	the Markov probability transition matrix for an Interacting Mixed Model (IMM) system
$\hat{\boldsymbol{x}}_j$, $\mathbf{P}_j$	the state estimate and covariance of the j-th filter in a multiple model system, or of the j-th target
$\hat{\boldsymbol{x}}^j$, $\mathbf{P}^j$	the modified state of the j-th model after mixing; provided as the input to the j-th filter in the IMM algorithm
s	the dimension of each measurement from a sensor (always 2 in this research)
$N_z(k)$	the number of measurements in the k-th set (i.e., the k-th scan)
N_t	the number of targets under track (one for this research)

Notation	Usage
$\theta_{ji}(k)$	a single measurement association event, indicating the association of measurement j with target i at sample k
$\Theta_l(k)$	the l-th joint association event for measurement set k, containing a single measurement event for each of the $N_m(k)$ measurements received in the k-th scan
$\psi_i(k)$	the i-th association event, proposing that the i-th measurement on the k-th scan is target-originated and the other $N_z(k)-1$ measurements are clutter-originated; the single-target case representation of $\theta_{ji}(k)$
$\mathbf{\Psi}_u(k)$	the u-th association *history* event, which contains a joint association event for each scan from 1 to k
$N_h(k)$	the number of association hypotheses in the tracking system after incorporation of the k-th set of measurements
$\mathbf{\Omega}_{N_h}(k)$	the full parameters (weights, means, covariances) of the N_h association hypotheses after incorporation of the k-th set of measurements
$N_r(k)$	the number of association hypotheses at the end of the k-th processing cycle, after hypothesis reduction has been applied
$\bar{\mathbf{\Omega}}_{N_r}(k)$	the parameters of the reduced set of N_r hypotheses

Multiple Model Adaptive Estimator Target Tracker For Maneuvering Targets in Clutter

I. Introduction

Modern remote target detection/tracking devices, such as radars, infrared sensor arrays, or electro-optical sensor arrays, can detect and resolve large numbers of objects within their field-of-view. Usually, only one or a few of these detected objects are of any practical interest, while the remaining objects are considerably less important or of no importance whatsoever. For example, an airborne radar on an Airborne Warning and Control System (AWACS) type of platform is only interested in airborne vehicles and is not interested in collateral objects such as birds, vehicles moving on the ground, trees, etc. These objects of no interest may, nonetheless, create radar returns that are equal to or greater in strength than the returns created by the airborne vehicles of interest. This being the case, even radars with substantial noise and multipath reduction may allow these returns to pass through the initial signal processing and be registered as *hypothetical targets*. The job of differentiating between which of these returns are of real interest and which ones are *clutter-originated* falls on information processing techniques implemented in tracker software.

Under the assumption that a real target (a fighter aircraft, for instance) does, in fact, exist within the sensor's field-of-view (its *measurement space*), some method must exist for keeping track of this target's current *state*. Usually, the target's state consists of a position, velocity, and possibly acceleration within the sensor's measurement space. Unfortunately, if the radar records multiple returns (known as *measurements*), there is considerable uncertainty about which return was produced by the assumed-to-exist target of interest and which were produced by clutter within the measurement space. The process of matching measurements with a real target of interest, known as *data association*, and the development of practical data association algorithms has received considerable attention by researchers.

The most modern algorithms rely upon simplified implementations of a Multiple Hypothesis Tracker (MHT) [22, 28], which, in its most fundamental form, records all possible permutations of hypothetical target-to-measurement matchings at every sensor scan. Each of these proposed matchings, referred to as *hypotheses*, are propagated forward in time on the next algorithm cycle using some assumed target dynamics model. At the next sensor scan, the propagated hypotheses are again matched to all incoming measurements, and the cycle starts over. Because these hypothesis-to-measurement matchings occur at every sensor scan, the system is actually maintaining a set of

measurement association histories, each of which represents a hypothetical target moving within the measurement space.

The goal of the MHT, therefore, is to *defer a decision* regarding which association histories are truly the most likely until more measurements have been recorded. As more measurements are recorded, clutter-originated measurements will tend to produce association histories (sometimes called *tracks*) that do not match the real target's *assumed* dynamics model. Conversely, measurements produced by the real target should generate tracks that closely adhere to the assumed model, so long as the assumed dynamics model is, indeed, correct for the target of interest. The process of deciding how well a track matches an assumed dynamics model relies on many of the same techniques used in conventional state estimation. The difference between the conventional state estimation problem and the tracking problem described here is that the former assumes a known and unchanging relationship exists between the state vector of interest and the measurements recorded by the sensor. Stated more succinctly, there is *no* measurement association uncertainty in the conventional state estimate problem.

Fundamental to this problem is the assumption that the i-th track's current state can be represented by a Gaussian probability density function, conditioned on that track's *assumed* measurement association history and *observed* measurement history, with a mean $\hat{\boldsymbol{x}}_i$, an error covariance $\mathbf{P}_i$, and a probability weight p_i. Each track from $i = 1$ to the number of existing tracks (or hypotheses), $N_h(k)$, represents a single *component* in a Gaussian *mixture* which is a probability-weighted sum of Gaussian densities. At any given time, the *entire mixture* or simplified approximation to the mixture, not necessarily any individual component, is the best representation available within the MHT of the true target's state. This is the essence of the deferred decision-making: as the number of measurement associations contained in each history grows, the total mixture will, ideally, be dominated by histories containing many associations with *target-originated* measurements because those components have high probability weights. Conversely, those histories containing many *clutter-originated* measurements will become increasingly less dominant because their probabilities diminish over time.

One characteristic of a MHT is that the number of components grow exponentially as time progresses, and some means must be used for limiting or reducing the number of *unlikely* components (i.e., those histories that are unlikely to represent a real target of interest) so that more likely components can continue to be matched with measurements on future cycles without a corresponding exponential growth in computational requirements. Considerable research has gone into developing reduction algorithms that properly optimize the merging and pruning choices in a mathematically tractable (and computationally efficient) manner [4, 5, 7, 13, 23]. Each of those techniques makes

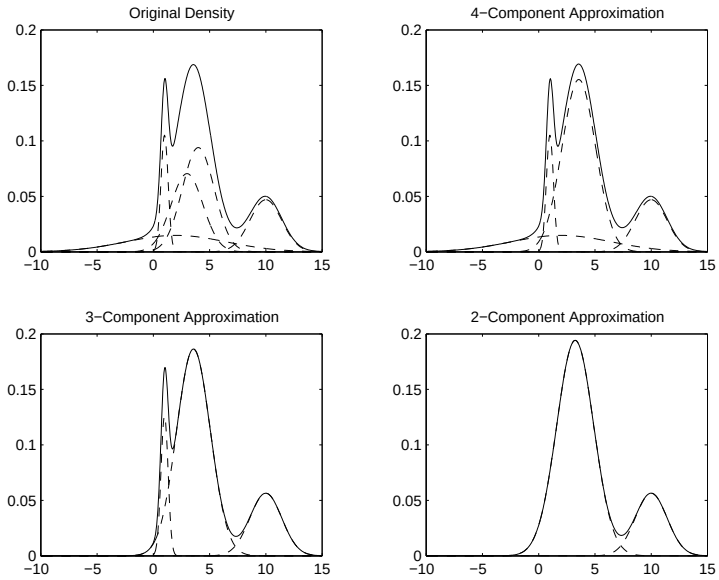

Figure 1.1: A conceptual example of Gaussian mixture reduction. The original mixture contains five components and the final post-reduction mixture contains fewer components. The solid black line is the probability-weighted sum (the mixture) of the individual components (plotted with dashed lines).

---

different assumptions about the tracking problem at hand and makes different simplifications to the concept of the Gaussian mixture representation.

Since the *complete mixture* best represents the true target state, an ideal reduction algorithm would attempt to maintain the general "shape" of the mixture while merging and pruning the actual components within the mixture. This process is illustrated in Figure 1.1. Fundamentally, the algorithm would assess a *cost* associated with merging two or more components or removing a component entirely. This cost represents the overall negative impact the action would have on the shape of the total mixture (i.e., how much worse does the lower-order mixture represent the pre-reduction, higher-order mixture). One such algorithm, developed by Williams [30], will be discussed in detail in Chapter 2 and subsequently applied to the tracking problems in this research.

The other problem facing any target tracking algorithm, whether or not measurement association uncertainty exists, is the fact that most targets of interest will exhibit poorly known dynamics. Returning to the AWACS example, the fighter aircraft of interest may be cruising over friendly territory at a constant velocity and constant altitude. Alternatively, the same aircraft flying over hostile territory could be maneuvering aggressively to avoid perceived threats from enemy radars, missiles, or aircraft. Whereas the former case might be best approximated by a constant-velocity dynamics model with a small amount of input driving noise at the acceleration level, the latter case might

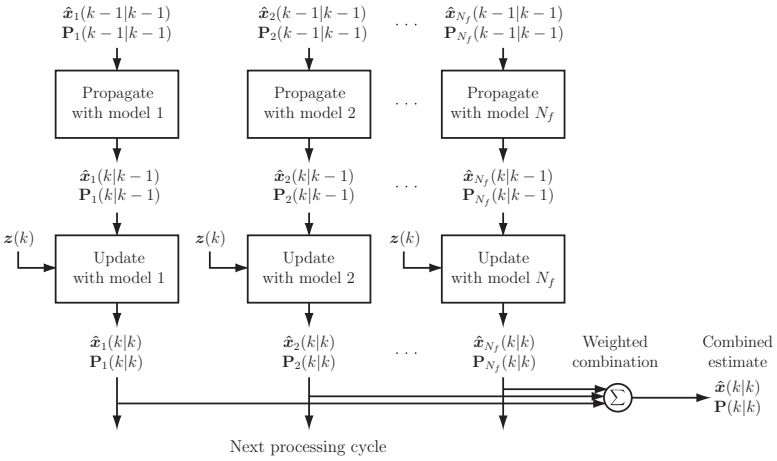

Figure 1.2: A fixed-model (MMAE) conceptual diagram in which each elemental filter assumes a unique dynamics model for its propagation cycle. In this research, the elemental filters will be Williams ISE mixture reduction filters instead of conventional Kalman filters. The incoming measurement set will be a *vector* of measurements, $\boldsymbol{Z}_k$, rather than a single measurement $\boldsymbol{z}(k)$ as depicted in the diagram.

---

be best approximated using a much more agile model – perhaps one involving large deterministic inputs at the velocity level or large amounts of input driving noise at the acceleration or jerk levels. Since the target tracking algorithm assumes a particular target dynamics model in its component propagation, some accounting must be made for any uncertainty in the dynamics model.

Several decades of research and application have shown that multiple-model adaptive estimation algorithms can satisfactorily address state estimation/target tracking problems when the state dynamics are poorly known [6,15,19]. Two examples of these algorithms, the Multiple-Model Adaptive Estimator (MMAE) and Interacting Mixed Model (IMM) estimator have received considerable attention in research publications and estimation theory texts [4, 17]. These techniques utilize individual estimation filters (typically Kalman filters) running in parallel, each assuming a different dynamics model, producing individual state estimates that are blended together probabilistically to form a single state estimate. Figure 1.2 shows a conceptual diagram of the static-model MMAE, but the concepts of elemental filters and filter estimate blending are applicable to both MMAE and IMM. In the research that follows, both forms will be discussed and compared in depth.

Most of the multiple-model algorithm research has focused on situations in which *no* measurement association uncertainty exists. Adapting these algorithms to cases with measurement association uncertainty is a relatively new area of interest, mainly because the computational demands imposed by modelling measurement association uncertainty *alongside* dynamics model uncertainty have been unrealistically high until recent years. Two different attempts to use IMM within an

MHT architecture are presented by Kirubarajan and Bar-Shalom [14] and by Dempster, Blackman, and Nichols [9]. Although both of these methods rely on the usual Bayesian assumptions inherent to the multiple-model algorithms, neither of them maintain Gaussian mixtures in the sense described above. Bar-Shalom uses the Probabilistic Data Association (PDA) method for *probabilistically* incorporating multiple measurements into a *single* track, so it is not an MHT algorithm. A PDA algorithm, in essence, forces the creation (at every measurement update cycle) of a single-component-approximation of the actual multi-component mixture that *would* exist in a true MHT algorithm. Blackman avoids the exponential growth of components in his proposal by allowing each incoming measurement to be associated with *only a single pre-existing component*. By putting a ceiling on the number of components spawned during measurement update, Blackman's approach, called an N-Scan filter, avoids the difficulties of mixture reduction entirely. Clearly, both of these methods make significant simplifications to the MHT philosophy in an attempt to integrate the multiple-model architecture.

1.1 Research Goal

The goal of this research is to combine the Integral Square Error (ISE) Gaussian mixture reduction algorithm developed by Williams with the MMAE and IMM multiple-model architecture. Each elemental filter of the multiple-model tracker will, itself, be a Williams filter instead of the typical Kalman filter. Each elemental Williams filter will maintain its *own* Gaussian mixture and will propagate the components of that mixture according to that elemental filter's assumed dynamics model. The estimates provided by each elemental Williams filter will be probabilistically combined in the same way the estimates for elemental Kalman filters are combined in a conventional MMAE or IMM. This will require the creation of a pseudo-state, pseudo-residual, and pseudo-error covariances using the Gaussian mixture components in place of the single state, residual, and associated error covariances provided by an elemental Kalman filter. Because there is some debate within the tracking community about the performance differences between MMAE and IMM, MHT trackers will be developed for both of these multiple-model algorithms.

Monte Carlo simulations will form the basis of all performance testing. The multiple-model algorithms will be tested in three different measurement clutter density environments, broadly representative of what is considered realistic for a modern tracking sensor such as a radar or infrared array. A host of filter bank configurations, each containing a different mix of filter-dynamics models, will be tested against a variety of simulated target truth models. The time-varying target truth models will be designed to replicate the behavior of benign and aggressively maneuvering airborne vehicles in such a way as to necessitate the use of a multiple-model approach (i.e., a single elemental filter using a single, time-invariant dynamics model would be insufficient for the tracking task). Finally, for

performance comparison, Kalman-filter-based equivalent algorithms will be run using the same filter bank configurations against the same truth model scenarios. The Kalman-filter-based algorithms will be run without measurement clutter, thereby eliminating any measurement association uncertainty, and this will form an asymptotic performance baseline for comparing Williams-filter-based algorithms as clutter density increases.

1.2 Assumptions

This research will assume one target exists in the measurement space with a known initial state at simulation start (with initial state uncertainty defined by an initial error covariance). Tracking problems will be limited to one target only, and the probability of detection (i.e., the probability that the target's true measurement exists within the complete set of measurements from each sensor scan) will be 1.0. Truth models will be piecewise-time-invariant, so changes in target dynamics will occur at predefined time instants. As a special subcase, a static truth dynamics models will remain in force for the entire interval. On certain truth models, target deterministic control inputs, like those exhibited by maneuvering airborne vehicles, will be allowed to change within an interval. All truth and filter design models are linear, stochastic models driven by zero-mean, white Gaussian noise and, depending on the model, may include deterministic control inputs.

The measurement model will assume a two-dimensional measurement space in Cartesian coordinates, broadly representative of a target trajectory in three-space *projected* onto a two-dimensional planar position sensor such as an infrared array or radar. It is assumed that range data is not necessary in the tracking problem of interest, and Cartesian coordinates are used to prevent the need for non-linear measurement models like that required for a sensor that generates azimuth/elevation data.

1.3 Thesis Organization

Chapter 2 provides background information on Kalman filtering (Section 2.2.1) and the general estimation problem assuming no measurement association uncertainty. Section 2.2.4 discusses the fixed-model MMAE architecture and Section 2.2.5 discusses switching-model algorithms with a primary emphasis on IMM. Section 2.3 develops the general concepts behind data association, Gaussian mixtures, and Gaussian mixture component reduction. Special emphasis is given the development of the ISE mixture reduction algorithm developed by Williams, and performance comparisons between ISE and alternative methods will be summarized.

Chapter 3 discusses the modifications necessary to replace the Kalman filters in the MMAE and IMM with elemental Williams filters. Section 3.2 discusses the use of partition-sensitive probabilities in systems for which elemental filters and truth models include dynamics models of differing state-dimensionality. Section 3.3 presents the creation of pseudo-states, pseudo-residuals, and pseudo-error covariances from the Williams filters that replace the states, residuals, and error covariances produced by conventional Kalman filters. Section 3.4 discusses quantitative means of determining how well an algorithm is tracking its target, and Section 3.5 covers the motivation behind the various *ad hoc* design choices necessary to implement these algorithms.

Chapter 4 presents Monte Carlo simulation results for the cases of no measurement clutter, "ultra-low"-clutter, low-clutter, and medium-clutter. Test results from various filter bank configurations running against various truth model scenarios are presented. Emphasis is placed on how well the multiple-model algorithms maintain track of the simulated targets executing a variety of maneuvers. Throughout the chapter, performance comparisons between equivalently configured MMAE and IMM architectures are examined. Chapter 5 concludes the research by summarizing the findings of Chapter 4 and suggesting further areas of research, with particular emphasis on algorithm optimization and tuning.

II. Background

2.1 Introduction

Target tracking problems have, historically speaking, been attacked using a wide array of techniques, all of which have merits and drawbacks when applied to specific scenarios. In a "real-world" application, two primary obstacles face the sensor/tracking algorithm package. The algorithm must maintain a target track in spite of the target's usually unknown and changing motion characteristics, while at the same time discern between real measurements (those produced by the target of interest) and false measurements (those produced by environmental clutter or targets of no interest).

Fortunately, for the designer, these problems can be handled separately to a degree. The first problem can be handled with a multiple model algorithm — either a Multiple Model Adaptive Estimator (MMAE) or Interacting Multiple Model (IMM) estimator — specifically designed to resolve the degree of maneuvering exhibited by the target. The second problem, assigning measurements to targets and discarding assignments likely to be clutter-originated, is handled by data association and hypothesis reduction algorithms. Although the two problems are treated separately during design, they interact in the final implementation to produce the overall target tracking solution.

Consequently, a high-performance data association/hypothesis reduction algorithm must be paired with a highly capable multiple-model tracking filter for the entire system to perform well. A realistic implementation might involve replacing the basic Kalman filters (which generally have no clutter rejection capability) as the elemental filters of this multiple model algorithm with filters designed specifically to work in conjunction with a clutter rejection algorithm. Section 2.1 will discuss tracking filter design with particular emphasis on multiple model techniques, and Section 2.2 will discuss data association and hypothesis reduction design.

2.2 Tracking Concepts

2.2.1 Kalman Filters. Traditionally, the Kalman Filter has formed the basis for a wide variety of solutions in the area of state estimation, failure detection, parameter identification, and target tracking. Kalman filter solutions are mathematically tractable, highly scalable in dimension, and highly tunable with respect to the physical parameters of interest. Consider a continuous-time, linear, time-varying system described by a state differential equation

$$\dot{\boldsymbol{x}}(t) = \mathbf{F}(t)\boldsymbol{x}(t) + \mathbf{B}(t)\boldsymbol{u}(t) + \boldsymbol{w}(t) \tag{2.1}$$

where $\boldsymbol{x}(t)$ is the state, $\boldsymbol{u}(t)$ is a vector of deterministic control inputs and $\boldsymbol{w}(t)$ is system dynamics driving noise described as white, Gaussian noise with mean and covariance kernel:

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}(t)\delta(t-t') \end{aligned} \tag{2.2}$$

The system has initial conditions $\boldsymbol{x}_0$, and since this is known imperfectly, has a mean $\hat{\boldsymbol{x}}_0$ and a covariance $\mathbf{P}_0$. At discrete time increments, measurements of the system's state will be taken. These measurements will be corrupted by discrete-time, zero-mean, white Gaussian noise such that

$$\boldsymbol{z}(t_k) = \mathbf{H}(t_k)\boldsymbol{x}(t_k) + \boldsymbol{v}(t_k) \tag{2.3}$$

where $\boldsymbol{v}(t_k)$ is assumed independent of $\boldsymbol{w}(t)$ and has statistics

$$\begin{aligned} E[\boldsymbol{v}(t_k)] &= \mathbf{0} \\ E[\boldsymbol{v}(t_k)\boldsymbol{v}^{\mathrm{T}}(t_j)] &= \mathbf{R}\delta_{jk} \end{aligned} \tag{2.4}$$

This continuous-time dynamics system, while correct, is not entirely useful for digital computer simulation and implementation. Instead, consider a linear, equivalent discrete-time system [18] with a *state-transition matrix* $\boldsymbol{\Phi}(k, k-1)$ that propagates the state from sample instant $k-1$ forward in time to instant k. Replacing the continuous time index t_k with the sample instant k, the state *difference* equation representation is

$$\boldsymbol{x}(k) = \boldsymbol{\Phi}(k, k-1)\boldsymbol{x}(k-1) + \mathbf{B_d}(k-1)\boldsymbol{u}(k-1) + \mathbf{w_d}(k-1) \tag{2.5}$$

where $\boldsymbol{u}(k)$ represents a discrete time control input, $\mathbf{w_d}(k)$ represents discrete-time, zero-mean, white Gaussian noise with statistics

$$\begin{aligned} E[\mathbf{w_d}(k)] &= \mathbf{0} \\ E[\mathbf{w_d}(k)\mathbf{w_d}^{\mathrm{T}}(l)] &= \mathbf{Q_d}\delta_{kl} \end{aligned} \tag{2.6}$$

and the system has initial conditions $\hat{\boldsymbol{x}}_0$ and $\mathbf{P}_0$. This system uses the same underlying physical dynamics as the continuous-time system, but by virtue of its discrete-time driving noise $\mathbf{w_d}$ and state-transition matrix $\boldsymbol{\Phi}(k, k-1)$, it is easily implemented on a digital computer.

The state dynamics matrices $\mathbf{\Phi}(k,k-1)$, $\mathbf{B_d}$, and $\mathbf{Q_d}$ are derived from their continuous time counterparts using the relationships:

$$\begin{array}{rclcl} \mathbf{\Phi}(k,k-1) & = & \mathbf{\Phi}(t_k,t_{k-1}) & = & e^{\mathbf{F}(t_k,t_{k-1})} \\ \mathbf{B_d}(k-1) & = & \mathbf{B_d}(t_{k-1}) & = & \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k,\tau)\mathbf{B}(\tau)d\tau \\ \mathbf{Q_d}(k-1) & = & \mathbf{Q_d}(t_{k-1}) & = & \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k,\tau)\mathbf{Q}(\tau)\mathbf{\Phi}^{\mathrm{T}}(t_k,\tau)d\tau \end{array} \tag{2.7}$$

where a time-invariant $\mathbf{F}$ matrix is assumed. For time-varying $\mathbf{F}(t)$, the state transition matrix would not be a matrix exponential, but the solution to $\dot{\mathbf{\Phi}}(t,t_{k-1}) = \mathbf{F}(t)\mathbf{\Phi}(t,t_{k-1})$ solved forward over a measurement sample period from the initial condition $\mathbf{\Phi}(t_{k-1},t_{k-1}) = \mathbf{I}$.

Given a linear, time-varying system, the optimal state estimator (optimal with respect to many criteria [18]) is the Kalman filter. Define the Kalman filter gain to be

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^{\mathrm{T}}(k)\left[\mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^{\mathrm{T}}(k) + \mathbf{R}(k)\right]^{-1} \tag{2.8}$$

where $\mathbf{P}(k|k-1)$ is the state estimate error covariance at instant k immediately *prior* to a measurement incorporation. The system incorporates measurements and then recalculates both the state estimate and its associated error covariance according to

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \mathbf{K}(k)\left[\boldsymbol{z}_k - \mathbf{H}(k)\hat{\boldsymbol{x}}(k|k-1)\right] \tag{2.9}$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}(k|k-1) \tag{2.10}$$

where $\hat{\boldsymbol{x}}(k|k)$ and $\mathbf{P}(k|k)$ represent the post-measurement state estimate and error covariance after the update. A dynamics propagation takes the system from its prior condition to a time immediately before a measurement update and is described by

$$\hat{\boldsymbol{x}}(k|k-1) = \mathbf{\Phi}(k,k-1)\hat{\boldsymbol{x}}(k-1|k-1) + \mathbf{B_d}(k-1)\boldsymbol{u}(k-1) \tag{2.11}$$

$$\mathbf{P}(k|k-1) = \mathbf{\Phi}(k,k-1)\mathbf{P}(k-1|k-1)\mathbf{\Phi}(k,k-1)^{\mathrm{T}} + \mathbf{Q_d}(k-1) \tag{2.12}$$

The so-called *linearized* Kalman filter and the *Extended Kalman Filter* (EKF) are analogous designs applied to non-linear systems. While they are not *optimal* filters for a non-linear system (in general, the optimal non-linear filter must be of infinite dimension), they are frequently used first-order approximate filters. Furthermore, compared to higher order non-linear filters, they are oftentimes more computationally feasible [17]. For the purposes of this research, the tracking will occur in a two-dimensional Cartesian space, and the corresponding dynamics equations are linear,

so linear models and linear Kalman filters will be emphasized. Some alternative target coordinate systems (azimuth/elevation angle, for example) would require one of the above linear approximations.

A major advantage of the Kalman filter is the fact that it is recursive. For a given system dynamics and measurement model, the state at the present time can be calculated using only the previous state estimate and the most recent measurement. It does not, therefore, require storing or batch processing of the complete measurement history, making it very efficient in real-time applications. [1] This is precisely the type of efficiency needed in an online, target tracking algorithm.

2.2.2 Target Truth Models. Typically, in a target tracking problem, the dynamics of the target of interest are never known with absolute, deterministic predictability. The degree of certainty about the target dynamics will greatly affect the tracker's performance, and the designer cannot ignore the possibility that target dynamics will vary over time. For time-varying target trajectory characteristics, one good approach is a *set* of models, with each model intended to depict a single *mode* of the target dynamics. Finding an appropriate $\mathbf{\Phi}$ matrix and $\mathbf{Q_d}$ matrix for each mode is the crux of the problem, and the solution could be approached in a number of ways.

Of particular interest in this research is an aerial target's trajectory in three-dimensional space projected onto a two-dimensional Cartesian plane. Such a projection would be realistic for tracking sensors such as radars, optical/IR seekers, etc., for which target range is not of immediate concern to the tracker. Commonly, aerial target trajectories are described using a first-order Gauss-Markov acceleration (FOGMA) model with *continuous-time* dynamics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{a}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \\ \dot{a}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ a_x(t) \\ y(t) \\ v_y(t) \\ a_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{T} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{T} \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{2.13}$$

where acceleration is a first-order (exponentially time-correlated) Gauss-Markov process, the output of a first-order lag (with lag coefficient $\frac{1}{T}$) driven by zero-mean, white, Gaussian noise with mean and covariance kernel

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\,\delta(t-t') \end{aligned} \tag{2.14}$$

[1] The measurement history is the collection of all measurement vectors from time zero to the present. The measurement history $\boldsymbol{Z}_k = [\boldsymbol{z}_1^{\mathrm{T}}\ \boldsymbol{z}_2^{\mathrm{T}}\ \cdots\ \boldsymbol{z}_k^{\mathrm{T}}]^{\mathrm{T}}$

where $\mathbf{Q} = q\mathbf{I}$, $q = 2T\sigma_a^2$, and σ_a is the rms acceleration level.

This model is physically motivated by the fact that, for many real-world targets, the power spectral density for acceleration is well approximated as roughly flat over a bandwidth of $\frac{1}{T}$ radians/sec. The magnitude of this acceleration is the noise strength q, and the time constant T determines roughly how long the vehicle maintains the current acceleration. For example, a highly maneuverable target like a missile might be characterized using large $\mathbf{Q}$ values and small T's, while a large transport aircraft would be better characterized using a small $\mathbf{Q}$ and large T.

To account for maneuvers (specifically, a change in target motion characteristics), two physically motivated and computationally straightforward modifications could be made to the above truth model. The most obvious change would be to change $\mathbf{Q}$ and T values over time. For example, a fighter aircraft in a high-altitude cruise would have small $\mathbf{Q}$ and large T. If the pilot senses illumination by a threat radar, a realistic modelling of evasive maneuvers might call for increasing $\mathbf{Q}$ and decreasing the T. Another way to handle such variations is to choose discrete point values of $\mathbf{Q}$ and T, one to represent each appropriate *mode* that such a target might display.

A physically realistic alternative to the FOGMA truth model might involve adding a non-zero $\mathbf{B_d}(k-1)$ matrix and a $\boldsymbol{u}(k-1)$ deterministic control input vector at the acceleration (thrust) level. Since "real-world" missiles and aircraft (piloted or otherwise) have control systems (either automatic or human) working to keep the vehicle's flight characteristics within certain parameters, a totally stochastic description of vehicle dynamics may be ignoring vital information about the target itself. Assuming the vehicle is, indeed, mostly controlled by aerodynamic forces, a thrust-perpendicular-to-velocity (TPV) model may be a good approximation to its motion characteristics. A TPV model accurately represents a constant g-load (thrust magnitude) turn in which the direction is intentionally kept perpendicular to the current velocity. The *continuous-time* TPV model is

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ y(t) \\ v_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_x & 0 \\ 0 & 0 \\ 0 & b_y \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \quad (2.15)$$

where the b terms are recalculated at each sample instant based on the current vehicle velocity magnitude and direction. Specifically, the b terms obey the relationships (in terms of speed s):

$$b_x = -v_y/s \quad (2.16)$$

$$b_y = v_x/s \quad (2.17)$$

$$s = \sqrt{v_x^2 + v_y^2} \quad (2.18)$$

In addition to the deterministic control inputs, acceleration is modelled by additive zero-mean, Gaussian, white-noise with mean and covariance kernel

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\,\delta(t-t') \end{aligned} \tag{2.19}$$

The best performing Kalman filter for target tracking applications with no measurement association uncertainty (i.e. no clutter) is a filter with state dynamics that exactly match those of the target and a measurement model that exactly matches the sensor characteristics. In most cases, the sensor model is known with much greater certainty than the target dynamics, so this research will assume measurement models are known and static in time, so the Kalman filter measurement matrix $\mathbf{H}$ and the measurement noise strength $\mathbf{R}$ are constant for all time. However, to extract maximum performance from the Kalman filter, it will be necessary to estimate the target dynamics online and adapt the Kalman filter accordingly. That is to say, the algorithm must estimate both the target states of interest (position, velocity, acceleration) *and* the dynamics parameters of interest ($\mathbf{\Phi}$, $\mathbf{Q_d}$, and $\mathbf{B_d}$). Methods of online estimation of dynamics parameters and adaptation of the Kalman filter equations will be discussed in the following sections.

2.2.3 Maximum Likelihood Estimators. One could use a maximum likelihood estimator (MLE) approach in which the designer chooses a likelihood function for maximization. This maximization would take the form

$$\left.\frac{\partial L[\boldsymbol{\theta}(k), \mathscr{Z}^k]}{\partial \boldsymbol{\theta}(k)}\right|_{\boldsymbol{\theta}(k)=\boldsymbol{\theta}^*(k)} = \mathbf{0}^{\mathrm{T}} \tag{2.20}$$

where $\boldsymbol{\theta}(k)$ represents a vector of unknown parameters (in this case values in the $\mathbf{\Phi}$, $\mathbf{B_d}$, and/or $\mathbf{Q_d}$ matrix) and $\mathscr{Z}^k$ represents a particular history of measurements collected up to the current instant k.

Specifically, for a target tracking problem, a designer may well be interested in a likelihood function such as [17]

$$\mathrm{L}\left[\boldsymbol{x}(k), \boldsymbol{a}, \boldsymbol{Z}^k\right] = \ln f_{\boldsymbol{x}(k),\boldsymbol{Z}^k|\boldsymbol{a}}(\boldsymbol{\xi}, \mathscr{Z}^k|\boldsymbol{\alpha}) \tag{2.21}$$

which is the natural log of the conditional probability density function of the state and measurement history up to the current time, conditioned on a particular set of model parameters $\boldsymbol{a}$ being in effect. Unfortunately, the maximum-likelihood approach typically requires numerical minimization using techniques like Newton-Raphson or Rao scoring. In a real-time target tracking application, the computational burden imposed by numerical minimization is a serious drawback.

2.2.4 Multiple Model Adaptive Estimation (MMAE). Rather than seeking an MLE solution, the designer might choose to approach the problem from a Bayesian standpoint. Consider the conditional density function relationship

$$f_{\boldsymbol{x}(k),\boldsymbol{a}|\boldsymbol{Z}^k}\left(\boldsymbol{\xi},\boldsymbol{\alpha}|\boldsymbol{\mathscr{Z}}^k\right) = f_{\boldsymbol{x}(k)|\boldsymbol{a},\boldsymbol{Z}^k}\left(\boldsymbol{\xi}|\boldsymbol{\alpha},\boldsymbol{\mathscr{Z}}^k\right) f_{\boldsymbol{a}|\boldsymbol{Z}^k}\left(\boldsymbol{\alpha}|\boldsymbol{\mathscr{Z}}^k\right) \tag{2.22}$$

where the term to the left of the equality is the joint density for the state and appropriate dynamics model parameters conditioned on a particular measurement history. To the right of the equality, the first term is the density of the state conditioned on the model parameters in effect and the measurement history. Note the parameter vector $\boldsymbol{a}$ is to right of the conditioning sign. That first term is of particular interest, because it is Gaussian with mean $\hat{\boldsymbol{x}}(k|k)$ and covariance $\mathbf{P}(k|k)$ as computed by a Kalman filter using the assumed dynamics model parameters $\boldsymbol{a} = \boldsymbol{\alpha}$.

Assume $\boldsymbol{a}$ can be any value in a continuous range $A \subset R^n$. The second term to the right of the equality in Equation (2.22) can be expressed by exploiting Bayes' rule and the concept of marginal densities as:

$$\begin{aligned} f_{\boldsymbol{a}|\boldsymbol{Z}^k}\left(\boldsymbol{\alpha}|\boldsymbol{\mathscr{Z}}^k\right) &= f_{\boldsymbol{a}|\boldsymbol{z}(k),\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\alpha}|\boldsymbol{\zeta}_k,\boldsymbol{\mathscr{Z}}^{k-1}\right) \\ &= \frac{f_{\boldsymbol{a},\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\alpha},\boldsymbol{\zeta}_k|\boldsymbol{\mathscr{Z}}^{k-1}\right)}{f_{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\zeta}_k|\boldsymbol{\mathscr{Z}}^{k-1}\right)} \\ &= \frac{f_{\boldsymbol{z}(k)|\boldsymbol{a},\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\zeta}_k|\boldsymbol{\alpha},\boldsymbol{\mathscr{Z}}^{k-1}\right) f_{\boldsymbol{a}|\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\alpha}|\boldsymbol{\mathscr{Z}}^{k-1}\right)}{\int_A f_{\boldsymbol{z}(k)|\boldsymbol{a},\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\zeta}_k|\boldsymbol{\alpha},\boldsymbol{\mathscr{Z}}^{k-1}\right) f_{\boldsymbol{a}|\boldsymbol{Z}^{k-1}}\left(\boldsymbol{\alpha}|\boldsymbol{\mathscr{Z}}^{k-1}\right) \mathrm{d}\boldsymbol{\alpha}} \end{aligned} \tag{2.23}$$

This expression is computationally infeasible in a real-time application, but if one restricts the parameter $\boldsymbol{a}$ to a finite set of values, the solution becomes substantially less burdensome. Consider a discretized parameter space and let $p_m(k)$ denote the probability that $\boldsymbol{a}$ will take on a discrete value $\boldsymbol{a}_m \in [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{N_f}]$. In such a space, $\boldsymbol{a}$ is a discrete random variable where each realization $\boldsymbol{a}_m$ either truly represents one of only N_f (so designated because this will be the number of elemental filters in the MMAE) possible values the parameter vector can assume, or is a discretized value meant to represent a set of possible continuous values the parameter might assume in the proximity of that discrete value. The end goal is a *proper discretization* of a continuous region of parameter space — a discretization that keeps the computational burden of the resulting estimator reasonable while maintaining satisfactory fidelity of the parameter vector estimate.

Define the conditional probability

$$p_m(k) \triangleq \mathrm{prob}\{\boldsymbol{a} = \boldsymbol{a}_m|\boldsymbol{Z}^k = \boldsymbol{\mathscr{Z}}^k\} \tag{2.24}$$

so that

$$f_{\boldsymbol{a}|\boldsymbol{Z}^k}\left(\boldsymbol{\alpha}|\mathscr{Z}^k\right) = \sum_{m=1}^{N_f} p_m(k)\delta(\boldsymbol{\alpha} - \boldsymbol{a}_m) \tag{2.25}$$

where

$$p_m(k) = \frac{f_{\boldsymbol{z}(k)|\boldsymbol{a},\boldsymbol{Z}^{k-1}}\left(\boldsymbol{z}_k|\boldsymbol{a}_m,\mathscr{Z}^{k-1}\right)\cdot p_m(k-1)}{\sum_{m=1}^{N_f} f_{\boldsymbol{z}(k)|\boldsymbol{a},\boldsymbol{Z}^{k-1}}\left(\boldsymbol{z}_k|\boldsymbol{a}_j,\mathscr{Z}^{k-1}\right)\cdot p_j(k-1)} \tag{2.26}$$

Notice that $p_m(k)$ is thus calculated recursively as a conditional density function multiplied by the probability coefficient from the previous time step and scaled by the sum of all possible numerator terms. The scaling imposed by the denominator ensures that $\sum_{m=1}^{N_f} p_m(k) \equiv 1$. The conditional density that appears in Eq. (2.26) is itself Gaussian

$$f_{\boldsymbol{z}(k)|\boldsymbol{a},\boldsymbol{Z}^{k-1}}\left(\boldsymbol{z}_k|\boldsymbol{a}_m,\mathscr{Z}^{k-1}\right) = \frac{1}{(2\pi)^{s/2}|\mathbf{A}_m(k)|^{1/2}}\exp\{\cdot\} \tag{2.27}$$

$$\{\cdot\} = \{-0.5\boldsymbol{r}_m^{\mathrm{T}}(k)\mathbf{A}_m^{-1}(k)\boldsymbol{r}_m(k)\} \tag{2.28}$$

$$\boldsymbol{r}_m(k) = \boldsymbol{z}(k) - \mathbf{H}_m(k)\hat{\boldsymbol{x}}_m(k|k-1) \tag{2.29}$$

$$\mathbf{A}_m(k) = \mathbf{H}_m(k)\mathbf{P}_m(k|k-1)\mathbf{H}_m^{\mathrm{T}}(k) + \mathbf{R}_m(k) \tag{2.30}$$

The measurement residual, $\boldsymbol{r}_m(k)$ is defined to be the difference between the actual measurement and the best prediction of that measurement from a Kalman filter based upon the assumption that $\boldsymbol{a}$ has assumed the value $\boldsymbol{a}_m$, and $\mathbf{A}_m(k)$ is the filter-computed covariance of that residual (see Figure 2.1).

Fundamentally, the MMAE is assigning a probability based on the magnitude of the measurement residual relative to its size anticipated by the filter-computed residual covariance $\mathbf{A}_m(k)$. Larger-than-anticipated residuals indicate the elemental filter is a poor representation of the actual target dynamics, and that filter gets a correspondingly small probability weight. Small residuals are created when the elemental filter is a good representation of the real world, and those filters receive higher weights. If the *best possible* model is used, the quadratic form $[\boldsymbol{r}_m^{\mathrm{T}}(k)\ \mathbf{A}_m^{-1}(k)\ \boldsymbol{r}_m(k)]$ in Eq. (2.28) should take on a value approximately equal to the measurement dimension s, while less appropriate models yield values that can be significantly larger than s.

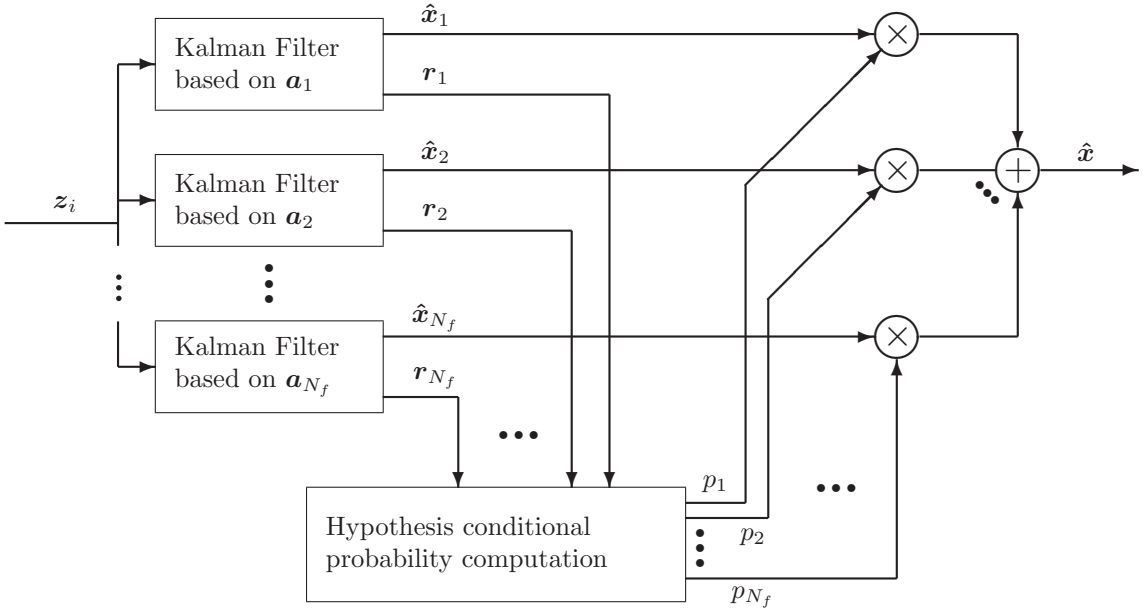

Figure 2.1: MMAE signal flow block diagram.

The MMAE-derived combined state estimate will be

$$\begin{aligned}
\hat{\boldsymbol{x}}(k|k) &= E\{\boldsymbol{x}(k)|\boldsymbol{Z}^k = \boldsymbol{\mathcal{Z}}^k\} \\
&= \int_{-\infty}^{\infty} \boldsymbol{\xi} \left[\sum_{m=1}^{N_f} f_{\boldsymbol{x}(k)|\boldsymbol{a},\boldsymbol{Z}^k}\left(\boldsymbol{\xi}|\boldsymbol{a}_m, \boldsymbol{\mathcal{Z}}^k\right) p_m(k) \right] \mathrm{d}\boldsymbol{\xi} \\
&= \sum_{m=1}^{N_f} \hat{\boldsymbol{x}}_m(k|k) p_m(k) \qquad (2.31)
\end{aligned}$$

Though generally not required, the covariance of this estimate can also be formed using a weighted average, but adding the correction term which takes into account the spreading introduced by the different estimates:

$$\mathbf{P}(k|k) = \sum_{m=1}^{N_f} p_m(k) \left\{ \mathbf{P}_m(k|k) + [\hat{\boldsymbol{x}}_m(k|k) - \hat{\boldsymbol{x}}(k|k)][\hat{\boldsymbol{x}}_m(k|k) - \hat{\boldsymbol{x}}(k|k)]^{\mathrm{T}} \right\} \qquad (2.32)$$

The resulting state estimate, therefore, can be expressed as a probability-weighted average of separate state estimates provided by corresponding Kalman filters, each of which assume a particular, discrete-valued dynamics model parameter vector [17] (see Figure 2.1). In addition, Eqs. (2.26) and (2.31) show the total solution up to the current time is the output of a recursive algorithm.

The model-conditioned state estimates and the model probability weights only require the current measurement and their own values at the previous time step for computation, rather than requiring knowledge of all measurements to the current time (as would be required by a batch estimator). This algorithm is known as multiple model adaptive estimation (MMAE) and has proven very effective in adaptive filtering, parameter/failure identification, adaptive control, and target tracking problems [10, 11, 16, 19, 20]. In this research, the dynamics model (state transition matrix $\mathbf{\Phi}$ values, input matrix $\mathbf{B_d}$, and driving noise covariance $\mathbf{Q_d}$) will be the primary elements of uncertainty.

Experience with MMAE systems has shown several modifications should be made to achieve higher levels of performance than what is achievable using the basic algorithm described above. The process of tuning an MMAE for maximum performance can be very application specific, although a few techniques have proven to be very useful in a wide spectrum of estimation problems. These techniques resolve certain difficulties inherent to the basic MMAE algorithm, and past research has shown them to be effective in elevating performance beyond that obtained by other adaptive methods without imposing significantly higher computational burden.

First and foremost, the designer must choose an appropriate discretization of the parameter space for the dynamics models. Research in this area has been extensive, and some applications have called for time-varying discretization (a so-called moving bank of filters) or time-varying number of filters (a growing or shrinking bank) [10, 11, 20]. In theory, these techniques are very applicable to target tracking problems, but this work will concentrate instead on maintaining a fixed number of static-parameter filters. For target tracking problems in which the targets are relatively far away from the sensor (an air traffic control radar, for example), this assumption is computationally efficient yet capable of handling a wide array of target types.

Secondly, it is necessary to establish a lower bound on the probability weight of a given model. Looking at Eq. (2.26), it is apparent that a probability weight equal to or near zero will be carried through many iterations before that model is again weighted sufficiently to be a realistic alternative to the other models in the bank. Lower bounding of elemental filter probabilities speeds this probability return by preventing any filter probabilities from approaching zero too closely.

One situation of special concern is the proper transition of models when the target undergoes a maneuver. At the onset of the maneuver, an MMAE's probability weight will quickly latch onto an aggressive dynamics filter, because the benign filter's measurement residual $\boldsymbol{r}_m$ becomes large very quickly compared to its computed covariance, $\mathbf{A}_m$ (which itself remains relatively small for benign dynamics filters). As the maneuver subsides, though, the benign filter, with its very low probability weight, experiences an undesirably slow return to its proper position as the most heavily weighted filter. More specifically, the high dynamics filter produces a large measurement residual

covariance $\mathbf{A}_m$, so obtaining a value of $[\boldsymbol{r}_m^{\mathrm{T}}(k)\ \mathbf{A}_m^{-1}(k)\ \boldsymbol{r}_m(k)]$ large enough to move the probability back towards the benign case is comparatively slow to occur. Korn and Beean [15] have published research on a restart mechanism that has proven quite effective in many MMAE implementations, although their technique proved unnecessary in this research.

Setting an upper bound on the quadratic form $[\boldsymbol{r}_m^{\mathrm{T}}(k)\ \mathbf{A}_m^{-1}(k)\ \boldsymbol{r}_m(k)]$, sometimes called the *filter divergence upper bound*, is a third *ad hoc* technique commonly used in MMAE design, and it has proven beneficial in dealing with situations like that described above. As an elemental filter becomes increasingly ill-suited to the current dynamics scenario, the actual measurement residual, $\boldsymbol{r}_m$, will become much larger than anticipated by the measurement residual covariance, $\mathbf{A}_m$, computed by the filter itself. Once this ratio becomes large enough, the filter is essentially diverging, and some means of resetting should be undertaken. Generally, a filter reset involves feedback of the MMAE blended estimate (the estimate composed of probability-weighted elemental filter estimates) to the divergent filter. Furthermore, the blended estimate itself is created using only the *non-divergent* filters with re-normalized probability weights such that their weights sum to one.[2]

2.2.5 Switching Models. The MMAE is a *fixed model* approach, in the sense that each elemental filter within the MMAE assumes that the parameter vector stays constant at $\boldsymbol{a}_m$ for all time. Model "switching" (i.e., changes in the probabilities that model $\boldsymbol{a}_m$ is in force at a particular sample instant) can only occur after a measurement update. Figure 2.2 shows the structure of the non-switching MMAE algorithm. The model-conditioned estimates calculated by each elemental filter at each processing cycle are passed directly into the same filter at the following processing cycle, as it is assumed that the model in force does not change with time. The overall combined estimate is calculated as a weighted average, as shown in Eqs. (2.31) and (2.32). The diagram in Figure 2.2 will serve as a basis of comparison for algorithms that do allow for model (mode) switching.

It can be seen from Equation (2.22) that the joint conditional probability density of the state $\boldsymbol{x}(k)$ and the dynamics model parameters $\boldsymbol{a}$ at sample instant k is conditioned upon the measurement history through k. In theory, the model that best represents a target's motion could change at any time, and algorithms sometimes referred to as *switching models* or *jump-linear systems* attempt to account for these changes that occur between measurement updates. A concise description of switching models appeared in a masters thesis by Williams [30], and much of the description that follows has been adapted from that text.

Each elemental filter in the MMAE was associated with a particular probability weighting, often referred to as a *mode probability*. This mode probability is defined as the probability of the

[2]These weights undergo yet another rescaling after the blended estimate feedback so that *all* elemental filters have probability weights that again sum to one.

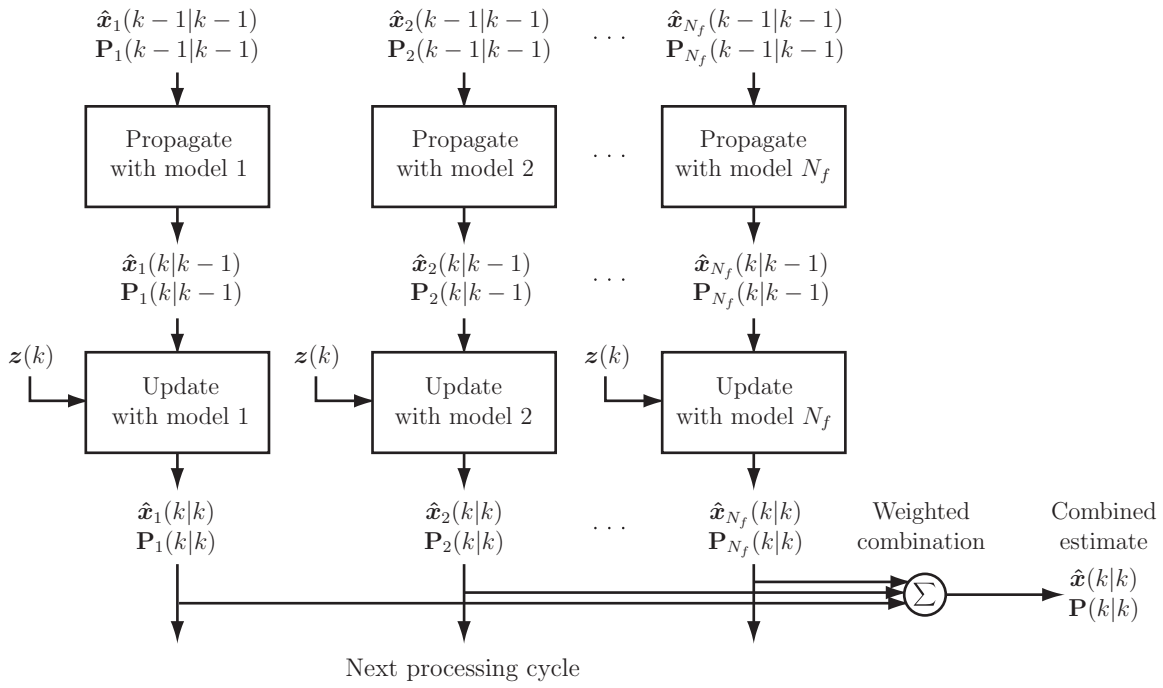

Figure 2.2: Block diagram of the non-switching Multiple Model Adaptive Estimator (MMAE).

parameter vector $\boldsymbol{a}$ assuming a discretized value $\boldsymbol{a}_m$ at instant k conditioned upon a particular measurement history $\boldsymbol{Z}^k = \boldsymbol{\mathcal{Z}}^k$ from sample instant 0 through k (see Equation (2.24)). Switching models account for the fact that any system of interest (in the tracking case, a target) has an associated *mode history* denoted as

$$\mathrm{M}^{k,l} = \{\mathrm{M}_{1,m1_l}, \mathrm{M}_{2,m2_l}, \ldots, \mathrm{M}_{k,mk_l}\} \tag{2.33}$$

The notation is interpreted as meaning that the l-th possible model history at time k consists of model m_{1_l} at sample time 1, model m_{2_l} at sample time 2, etc., where each m_{j_l} is the index to a model number between 1 and the number of elemental filters, N_f. If transitions are allowed to any of the N_f models at any sample instant, then every model history event at time k will give rise to N_f new events at time $(k+1)$, and so the number of possible model histories increases exponentially with time according to ${N_f}^k$. The PDF of the target state conditioned on the measurements must then be calculated as a total probability expansion over all model history events:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{l=1}^{{N_f}^k} f\{\boldsymbol{x}(k)|M^{k,l}, \boldsymbol{Z}^k\} P\{M^{k,l}|\boldsymbol{Z}^k\} \tag{2.34}$$

The model history probability $P\{M^{k,l}|\boldsymbol{Z}^k\}$ is expanded as:

$$\begin{aligned} P\{M^{k,l}|\boldsymbol{Z}^k\} &= P\{M^{k,l}|\boldsymbol{z}(k),\boldsymbol{Z}^{k-1}\} \\ &= \frac{f\{M^{k,l},\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \\ &= \frac{f\{\boldsymbol{z}(k)|M^{k,l},\boldsymbol{Z}^{k-1}\}P\{M^{k,l}|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \\ &= \frac{f\{\boldsymbol{z}(k)|M^{k,l},\boldsymbol{Z}^{k-1}\}P\{M_{k,j},M^{k-1,l'}|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \\ &= \frac{f\{\boldsymbol{z}(k)|M^{k,l},\boldsymbol{Z}^{k-1}\}P\{M_{k,j}|M^{k-1,l'},\boldsymbol{Z}^{k-1}\}P\{M^{k-1,l'}|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \end{aligned} \tag{2.35}$$

where l is the index of the current model history (between 1 and ${N_f}^k$), l' is the index of the previous model history (between 1 and ${N_f}^{k-1}$), and j is the index of the current model (between 1 and N_f) hypothesized by the model history event $M^{k,l}$. The denominator is expanded as a total probability expansion over all model history events as in Eq. (2.34).

The method commonly used to evaluate model history event probabilities is to assume that the model transition process is a Markov process represented by a probability transition matrix $\mathbf{T}(k|k-1)$ such that

$$\begin{bmatrix} P\{M_{k,1}\} \\ P\{M_{k,2}\} \\ \vdots \\ P\{M_{k,N_f}\} \end{bmatrix} = \mathbf{T}(k|k-1) \begin{bmatrix} P\{M_{k-1,1}\} \\ P\{M_{k-1,2}\} \\ \vdots \\ P\{M_{k-1,N_f}\} \end{bmatrix} \tag{2.36}$$

where

$$\mathbf{T}(k|k-1) = \begin{bmatrix} p_{1,1}(k) & p_{2,1}(k) & \cdots & p_{N_f,1}(k) \\ p_{1,2}(k) & p_{2,2}(k) & \cdots & p_{N_f,2}(k) \\ \vdots & \vdots & \vdots & \vdots \\ p_{1,N_f}(k) & p_{2,N_f}(k) & \cdots & p_{N_f,N_f}(k) \end{bmatrix} \tag{2.37}$$

and the probability of transition depends *only* on the previous model number $m_{k-1_{l'}}$, and not on the prior model history or prior measurements:

$$\begin{aligned} P\{M_{k,j}|M^{k-1,l'},\boldsymbol{Z}^{k-1}\} &= P\{M_{k,j}|M^{k-1,l'}\} \\ &= P\{M_{k,j}|M_{1,m_{1_{l'}}},M_{2,m_{2_{l'}}},\ldots M_{k-1,m_{k-1_{l'}}}\} \\ &= P\{M_{k,j}|M_{k-1,m_{k-1_{l'}}}\} \\ &\triangleq p_{m_{k-1_{l'}}j}(k) \end{aligned} \tag{2.38}$$

Thus $p_{m_{k-1_{l'}}j}(k)$ is the probability of transitioning from model index $m_{k-1_{l'}}$ at sample time $(k-1)$ to model index j at sample time k, where each index is a model number between 1 and N_f. The choice of values of the elements for $\mathbf{T}(k|k-1)$ relies largely on insight into the particular estimation problem at hand, although for most problems involving physical vehicles (target tracking being a good example), this matrix is diagonally dominant. Chapter 3 will discuss in more detail the choices made in specifying the Markov probability transition matrices for this research. Note that each column of $\mathbf{T}(k|k-1)$ must sum to one, since the total probability of reaching *any* of the admissible modes from a given mode must equal unity. For this research, the elements of $\mathbf{T}(k|k-1)$ are assumed to be constant for all time.

While the assumption of Eq. (2.38) provides a mechanism for computation of the model history probability, the conditioning in Eq. (2.35) of the new measurement probability on the model history still produces an exponentially increasing number of hypotheses with time, and so further approximation (such as combining branches) is required. The most commonly used algorithms are described in the following sections. The structure of the full order Bayesian switching estimator is shown in Figure 2.3. The diagram demonstrates the growing number of filters which is required: the output of every filter at the current processing cycle must be processed in the following processing cycle using every model, hence the number of filtering operations at the k-th cycle is ${N_f}^k$.

2.2.6 First-Order Generalized Pseudo-Bayesian Estimator. The First-Order Generalized Pseudo-Bayesian (GPB-1) estimator [3, 30] limits the memory of the model history events by combining the estimates from all models into a single estimate at the end of each processing cycle. At the start of each processing cycle, the information carried forward from the previous measurement interval is a single combined estimate: any conditioning on previous model history events has been discarded. Hence the PDF of the estimate is modified from the switching model of Eq. (2.34):

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{l=1}^{{N_f}^k} f\{\boldsymbol{x}(k)|M^{k,l}, \boldsymbol{Z}^k\} P\{M^{k,l}|\boldsymbol{Z}^k\}$$

to the simplified version:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k,j}|\boldsymbol{Z}^k\} \tag{2.39}$$

where the total probability expansion over the entire model history event $M^{k,l}$ is replaced by expansion over the single most recent model event $M_{k,j}$. Expanding Eq. (2.39), we further approximate that the previous measurement history $\boldsymbol{Z}^{k-1}$ is adequately represented by the *single* estimate and

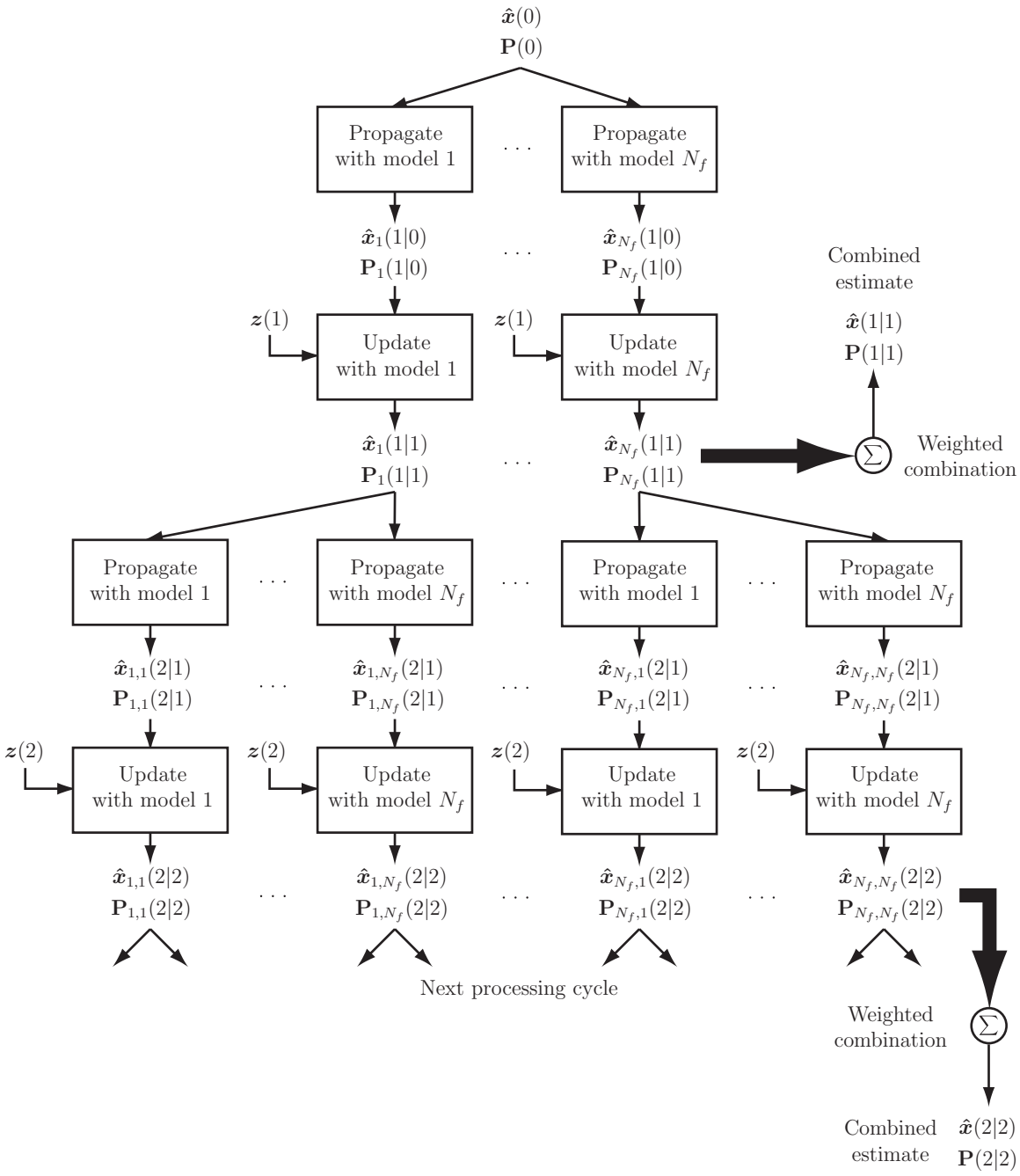

Figure 2.3: Block diagram of full order Markov switching estimator.

covariance from the previous processing cycle, $\{\hat{\boldsymbol{x}}(k-1|k-1), \mathbf{P}(k-1|k-1)\}$:

$$\begin{aligned} f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} &= \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k,j}|\boldsymbol{Z}^k\} \\ &= \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j}, \boldsymbol{z}(k), \boldsymbol{Z}^{k-1}\} P\{M_{k,j}|\boldsymbol{Z}^k\} \\ &= \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j}, \boldsymbol{z}(k), \hat{\boldsymbol{x}}(k-1|k-1), \mathbf{P}(k-1|k-1)\} \cdot \\ &\qquad \cdot P\{M_{k,j}|\boldsymbol{Z}^k\} \end{aligned} \tag{2.40}$$

In effect, the approximations of Eq. (2.39) and (2.40) mean to say that the entire model transition history and measurement history are representable through the single estimate from the previous processing cycle. Once the conditional model probability in Eq. (2.40) has been evaluated using the developments of Eq. (2.35) and (2.38), the combined estimate is then calculated as per Eqs. (2.31) and (2.32) and the cycle repeats.

The structure of the GPB-1 algorithm is shown in Figure 2.4. The outputs of all filters are merged into a single estimate at each processing cycle, which is used as the input to each of the filters at the next processing cycle, providing a very coarse approximation of the optimal system shown in Figure 2.3.

2.2.7 Second-Order Generalized Pseudo-Bayesian Estimator. The Second-Order Generalized Pseudo-Bayesian (GPB-2) estimator [3, 30] operates on similar principles to the first-order variant, except that the memory is allowed to extend an additional processing cycle. Again, the PDF of the estimate is modified from the full order switching model of Eq. (2.34):

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{l=1}^{{N_f}^k} f\{\boldsymbol{x}(k)|M^{k,l}, \boldsymbol{Z}^k\} P\{M^{k,l}|\boldsymbol{Z}^k\}$$

to the simplified version, which this time incorporates the *previous* model $M_{k-1,i}$ in addition to the current model $M_{k,j}$:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k-1,i}, M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k-1,i}, M_{k,j}|\boldsymbol{Z}^k\} \tag{2.41}$$

Manipulating Eq. (2.41) and assuming that the history $\{M_{k-1,i}, \boldsymbol{Z}^{k-1}\}$ is adequately represented by the combined estimates from the i-th model in the previous processing cycle $\{\hat{\boldsymbol{x}}_i(k-1|k-1), \mathbf{P}_i(k-1|k-1)\}$, and that (according to the Markov model) the model transition

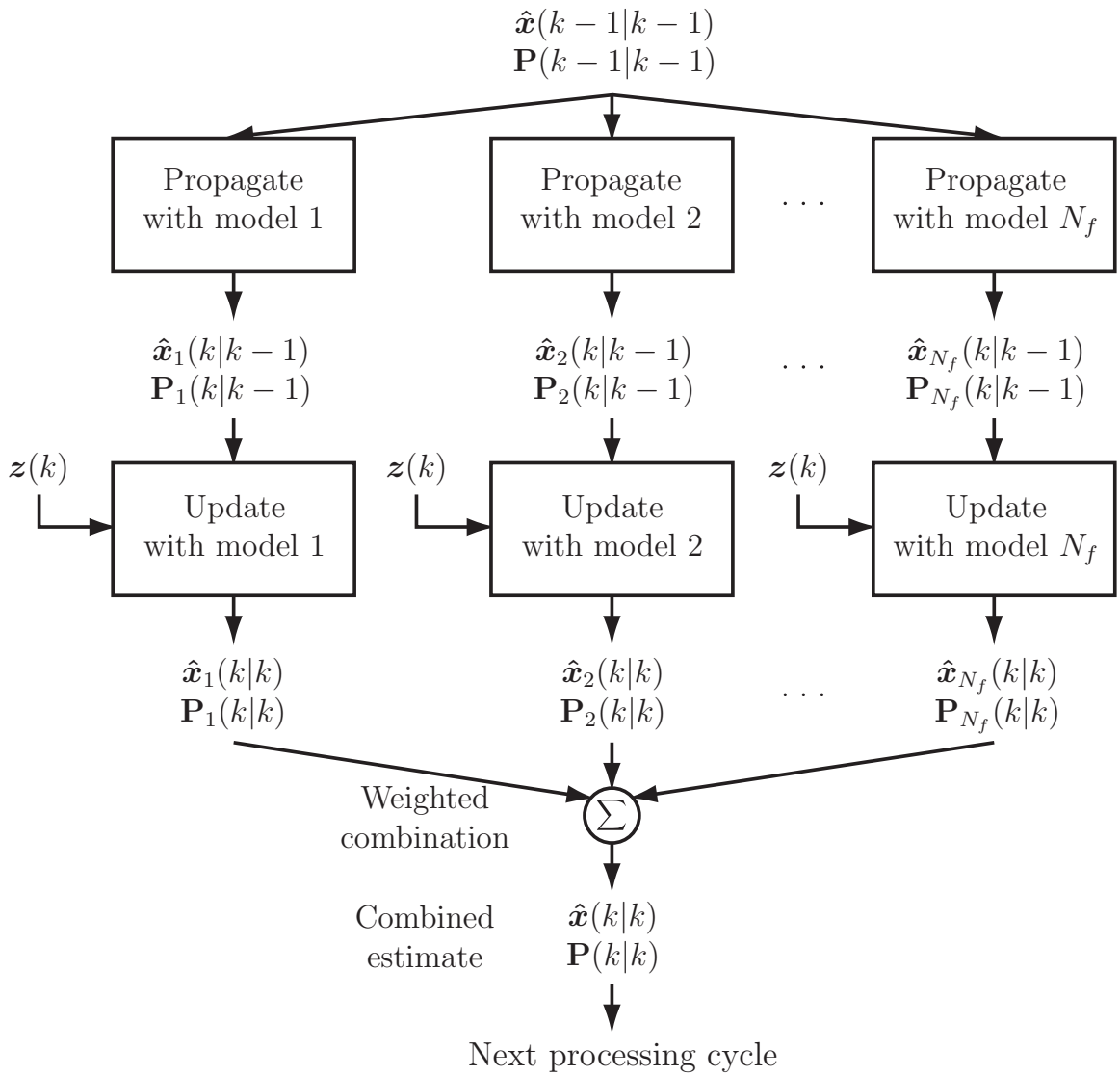

Figure 2.4: Block diagram of GPB-1 algorithm.

depends only on the previous model, and not on the measurement history:

$$\begin{aligned} f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} &= \sum_{i=1}^{N_f}\sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j},\boldsymbol{z}(k),\{M_{k-1,i},\boldsymbol{Z}^{k-1}\}\}\cdot \\ &\qquad \cdot P\{M_{k,j}|M_{k-1,i},\boldsymbol{Z}^k\}P\{M_{k-1,i}|\boldsymbol{Z}^k\} \\ &= \sum_{i=1}^{N_f}\sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j},\boldsymbol{z}(k),\hat{\boldsymbol{x}}_i(k-1|k-1),\mathbf{P}_i(k-1|k-1)\}\cdot \\ &\qquad \cdot P\{M_{k,j}|M_{k-1,i}\}P\{M_{k-1,i}|\boldsymbol{Z}^k\} \end{aligned} \tag{2.42}$$

Therefore, the operation of the GPB-2 algorithm is such that the estimate from each model in the previous processing cycle is processed using each dynamics model, giving ${N_f}^2$ total elemental filters. At the end of each processing cycle, the ${N_f}^2$ estimates are combined down to N_f estimates, combining estimates from different models in the *previous* processing cycle to leave *one estimate* for each model in the *latest* processing cycle. This is illustrated in Figure 2.5, which shows the structure of the algorithm. Comparing the structure to the GPB-1 algorithm shown in Figure 2.4, the GPB-2 algorithm uses ${N_f}^2$ filters, thus it is able to maintain N_f estimates and propagate each estimate with each of the N_f filters at each processing interval, rather than collapsing the PDF of target state down to a single estimate at each processing interval.

2.2.8 Interacting Multiple Model Estimator. The Interacting Multiple Model (IMM) estimator [3, 6, 30] is a methodology which achieves comparable performance to the GPB-2 estimator using only N_f elemental filters, rather than ${N_f}^2$ as required by the latter. The algorithm can be derived by considering the limitations inherent to the problem: if only N_f elemental filters are allowable, then the input to the j-th filter should be the best estimate of the state at time instant $(k-1)$, conditioned on the event that model j is in force *at time instant* k (the new sample time), $f\{\boldsymbol{x}(k-1)|M_{k,j},\boldsymbol{Z}^{k-1}\}$. Using this expression as a starting point, we follow a single iteration of the algorithm, through to the calculation of the same function at the following sample period.

Following a standard Kalman filter propagate-update cycle at the k-th sample time, the output of the j-th elemental filter will be $f\{\boldsymbol{x}(k)|M_{k,j},\boldsymbol{Z}^k\}$. The requirement for the IMM algorithm is thus to combine the estimates from the N_f elemental filters to calculate the inputs $f\{\boldsymbol{x}(k)|M_{k+1,i},\boldsymbol{Z}^k\}$ of each elemental filter for the next processing cycle.

The overall PDF formed using the information from all N_f filters represents the total information contained by the system at time k:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k,j},\boldsymbol{Z}^k\}P\{M_{k,j}|\boldsymbol{Z}^k\} \tag{2.43}$$

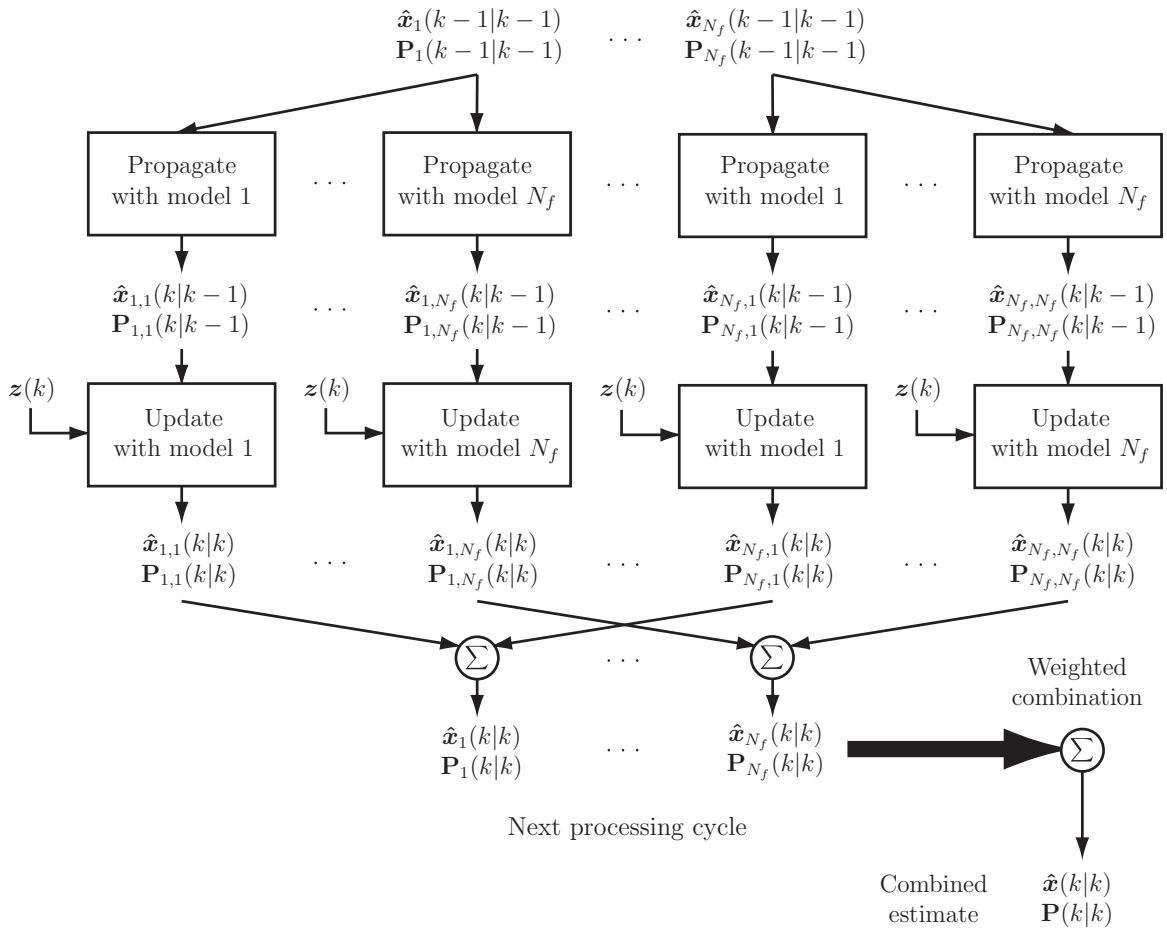

Figure 2.5: Block diagram of GPB-2 algorithm.

The goal of the intermixing is thus to massage Eq. (2.43) into the expansion necessary at the input to the next processing cycle:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{i=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k+1,i}, \boldsymbol{Z}^k\} P\{M_{k+1,i}|\boldsymbol{Z}^k\} \tag{2.44}$$

The last factor in Eq. (2.44) is easily evaluated using the Markov assumption as per Eq. (2.38):

$$\begin{aligned} P\{M_{k+1,i}|\boldsymbol{Z}^k\} &= \sum_{j=1}^{N_f} P\{M_{k+1,i}|M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k,j}|\boldsymbol{Z}^k\} \\ &= \sum_{j=1}^{N_f} P\{M_{k+1,i}|M_{k,j}\} P\{M_{k,j}|\boldsymbol{Z}^k\} \end{aligned} \tag{2.45}$$

Note that if $\mathbf{T}(k+1|k)$ is the Markov transition matrix such that:

$$\{\mathbf{T}\}_{ij} = P\{M_{k+1,i}|M_{k,j}\}$$

then Eq. (2.45) is simply a matrix multiplication of $\mathbf{T}(k+1|k)$ by the vector with elements that are the probabilities $P\{M_{k,j}|\boldsymbol{Z}^k\}$, yielding the vector of components that represent the probabilities $P\{M_{k+1,i}|\boldsymbol{Z}^k\}$.

The leading factor in the sum of Eq. (2.44) is then expanded using the total probability theorem over the previous model index j:

$$f\{\boldsymbol{x}(k)|M_{k+1,i}, \boldsymbol{Z}^k\} = \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k+1,i}, M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\} \tag{2.46}$$

where the *backward* transition probabilities (sometimes called the *mixing probabilities*) are calculated by:

$$\begin{aligned} P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\} &= \frac{P\{M_{k,j}, M_{k+1,i}|\boldsymbol{Z}^k\}}{P\{M_{k+1,i}|\boldsymbol{Z}^k\}} \\ &= \frac{P\{M_{k+1,i}|M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k,j}|\boldsymbol{Z}^k\}}{P\{M_{k+1,i}|\boldsymbol{Z}^k\}} \\ &= \frac{P\{M_{k+1,i}|M_{k,j}, \boldsymbol{Z}^k\} P\{M_{k,j}|\boldsymbol{Z}^k\}}{\sum_{n=1}^{N_f} P\{M_{k+1,i}|M_{k,n}, \boldsymbol{Z}^k\} P\{M_{k,n}|\boldsymbol{Z}^k\}} \end{aligned} \tag{2.47}$$

According to the Markov assumption, the transition probability $P\{M_{k+1,i}|M_{k,j}, \boldsymbol{Z}^k\}$ does not depend on the measurement history $\boldsymbol{Z}^k$, hence this conditioning is dropped.

Assuming that the estimator history $\boldsymbol{Z}^k$ is adequately modelled by the N_f estimates from the previous processing cycle (each estimate conditioned on a different model $M_{k,j}$), Eq. (2.46) is then approximated by a single Gaussian density:[3]

$$\begin{aligned} f\{\boldsymbol{x}(k)|M_{k+1,i},\boldsymbol{Z}^k\} &\approx \sum_{j=1}^{N_f} f\{\boldsymbol{x}(k)|M_{k+1,i},\hat{\boldsymbol{x}}_j(k|k),\mathbf{P}_j(k|k)\}P\{M_{k,j}|M_{k+1,i},\boldsymbol{Z}^k\} \\ &\approx \mathcal{N}\{\boldsymbol{x}(k);\hat{\boldsymbol{x}}^i(k|k),\mathbf{P}^i(k|k)\} \end{aligned} \tag{2.48}$$

where the mean and variance of the Gaussian are given by:

$$\hat{\boldsymbol{x}}^i(k|k) = \sum_{j=1}^{N_f} P\{M_{k,j}|M_{k+1,i},\boldsymbol{Z}^k\}\hat{\boldsymbol{x}}_j(k|k) \tag{2.49}$$

$$\begin{aligned} \mathbf{P}^i(k|k) = \sum_{j=1}^{N_f} P\{M_{k,j}|M_{k+1,i},\boldsymbol{Z}^k\}\{\mathbf{P}_j(k|k) + \\ + [\hat{\boldsymbol{x}}_j(k|k)-\hat{\boldsymbol{x}}^i(k|k)][\hat{\boldsymbol{x}}_j(k|k)-\hat{\boldsymbol{x}}^i(k|k)]^T\} \end{aligned} \tag{2.50}$$

The *a posteriori* model probabilities $P\{M_{k,j}|\boldsymbol{Z}^k\}$ (also called the *mode probabilities*) required for Eq. (2.47) are calculated recursively using the expressions:

$$\begin{aligned} P\{M_{k,j}|\boldsymbol{Z}^k\} &= P\{M_{k,j}|\boldsymbol{z}(k),\boldsymbol{Z}^{k-1}\} \\ &= \frac{f\{M_{k,j},\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \\ &= \frac{f\{\boldsymbol{z}(k)|M_{k,j},\boldsymbol{Z}^{k-1}\}P\{M_{k,j}|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \end{aligned} \tag{2.51}$$

As discussed in Eq. (2.45), $P\{M_{k,j}|\boldsymbol{Z}^{k-1}\}$ can be expanded using the total probability theorem as:

$$\begin{aligned} P\{M_{k,j}|\boldsymbol{Z}^{k-1}\} &= \sum_{i=1}^{N_f} P\{M_{k,j}|M_{k-1,i},\boldsymbol{Z}^{k-1}\}P\{M_{k-1,i}|\boldsymbol{Z}^{k-1}\} \\ &= \sum_{i=1}^{N_f} P\{M_{k,j}|M_{k-1,i}\}P\{M_{k-1,i}|\boldsymbol{Z}^{k-1}\} \end{aligned} \tag{2.52}$$

where the assumption that the model transition probability does not depend on the measurement history is again invoked.

[3]Note that $\{\hat{\boldsymbol{x}}_j(k|k),\mathbf{P}_j(k|k)\}$ is taken to refer to the filter estimate at the output of the previous processing cycle, while $\{\hat{\boldsymbol{x}}^i(k|k),\mathbf{P}^i(k|k)\}$ represents the *mixed* estimates to be provided at the input to the next processing cycle.

Thus by substituting in Eq. (2.52) and expanding the denominator using the total probability theorem, Eq. (2.51) becomes:

$$P\{M_{k,j}|\boldsymbol{Z}^k\} = \frac{f\{\boldsymbol{z}(k)|M_{k,j},\boldsymbol{Z}^{k-1}\}\sum_{i=1}^{N_f} P\{M_{k,j}|M_{k-1,i}\}P\{M_{k-1,i}|\boldsymbol{Z}^{k-1}\}}{\sum_{n=1}^{N_f} f\{\boldsymbol{z}(k)|M_{k,n},\boldsymbol{Z}^{k-1}\}P\{M_{k,n}|\boldsymbol{Z}^{k-1}\}} \tag{2.53}$$

(Note that the denominator is simply the scaling factor necessary to ensure that the conditional model probabilities sum to unity.)

As per the preceding multiple model techniques, the combined estimate is calculated at each processing cycle to give the output of the estimator. A block diagram of the IMM algorithm is shown in Figure 2.6. The structure is very similar to the non-switching MMAE structure shown in Figure 2.2: there are N_f filters, each of which is supplied with a different input. However, rather than passing the output of each filter directly into the same filter at the next processing cycle, the algorithm mixes the estimates according to the Markov transition model in order to allow the system to react to changes to the model in force.

2.2.9 Summary. Of particular interest to this research is the performance of the fixed-model MMAE and the switching-model IMM, because experience has shown these two algorithms provide a good compromise between blended estimate fidelity and computational efficiency in many applications. A brief summary of these two techniques will hopefully emphasize the major differences between the two as well as illuminate some areas worth examining further.

The traditional MMAE is based on the assumption that the model in force does not change with time; *ad hoc* modifications extend the algorithm to provide adequate performance in a switching model environment. Furthermore, the MMAE assumes the model probabilities can only change after a measurement cycle, evidenced in the fact that the mode probability conditional density in Eq. (2.22), $f_{\boldsymbol{a}|\boldsymbol{Z}^k}\left(\boldsymbol{\alpha}|\boldsymbol{\mathcal{Z}}^k\right)$ is conditioned upon the measurement history through instant k. The blended estimate is calculated as a weighted blending of elemental filter states and covariances according to Eqs. (2.31) and (2.32) with weighting provided by the calculated mode probabilities. These mode probabilities are computed recursively according to Eq. (2.26), and they depend only on the elemental filter's measurement residual conditional PDF and the mode probabilities at the previous time instant $k-1$. Using *ad hoc* techniques, elemental filters with diverging solutions may be "restarted" using the blended state and covariance as Kalman filter initial conditions. Also, elemental filters may have a mode probability lower bound imposed to improve the speed of MMAE response when the target is initiating or ceasing a maneuver.

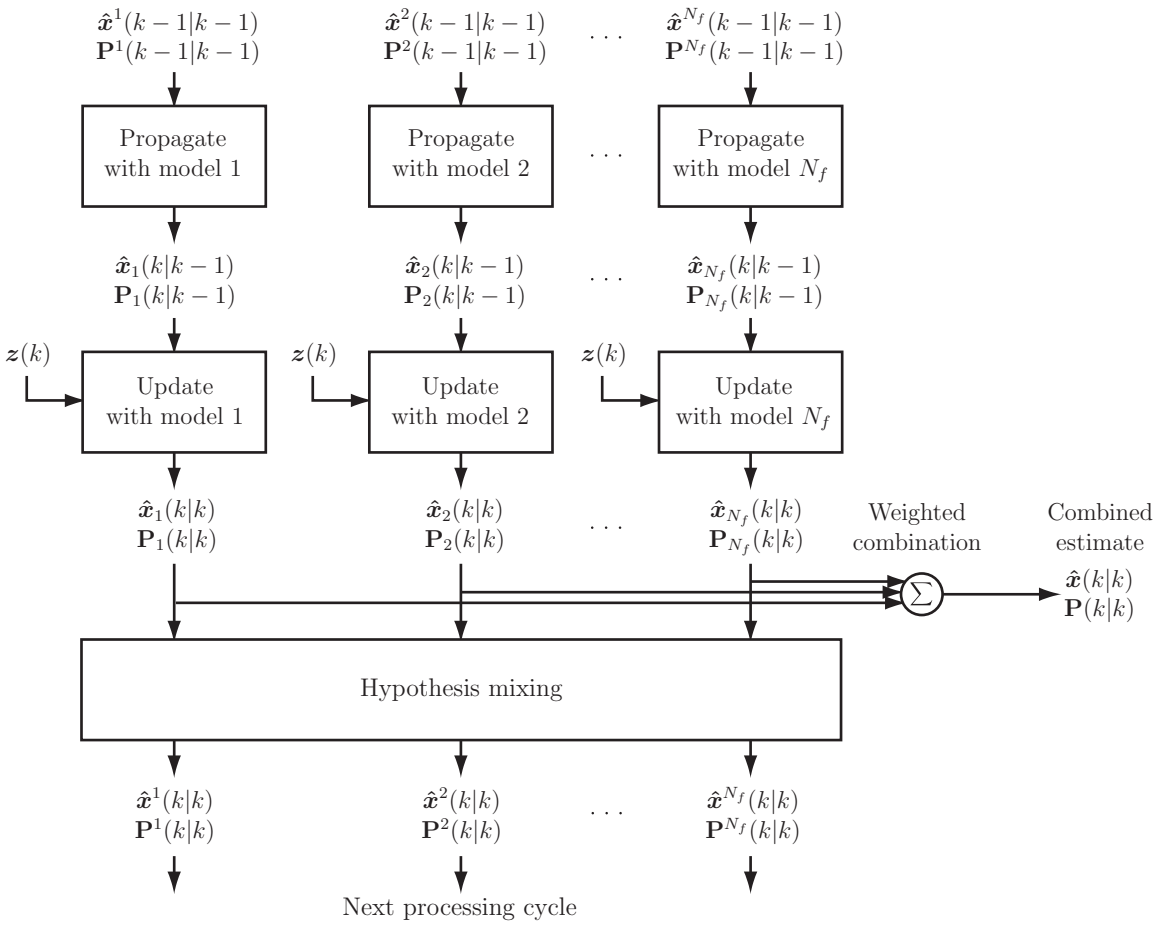

Figure 2.6: Block diagram of IMM algorithm.

IMM, a switching-model algorithm, adds a Markov assumption to model transitions *between measurement updates*. Thus, one can conceive of a "propagation cycle" as well as a "measurement update cycle" for the mode probabilities. The essence of this Markov assumption is that mode probabilities from sample instant $k-1$ migrate towards different values according to a Markov probability transition matrix $\mathbf{T}(k|k-1)$ as shown in Eq. (2.36) and, therefore, themselves behave as a random process. The values within $\mathbf{T}$ are chosen using insight into the tracking problem of interest. An IMM's blended output has a conditional PDF according to Eq. (2.43), where the probability weighting terms, much like MMAE, are the mode probabilities at instant k. Mode probabilities in IMM are computed similarly to their calculation within the MMAE except that the terms in $\mathbf{T}$ appear as additional weighting factors. Finally, each elemental filter receives a new initial condition with conditional PDF according to Eq. (2.46) and as implemented via Eqs. (2.49) and (2.50). This mixing occurs every processing cycle and, in theory, eliminates the need for the *ad hoc* techniques used in MMAE to address changing modes.

A fundamental connection between MMAE and IMM exists when the IMM's Markov transition matrix is an identity. Note that, if the Markov transition matrix defined below Eq. (2.45) is an identity matrix, this would correspond to the static-parameter assumption inherent in the conventional MMAE. Under this condition, the mixing probabilities in Eq. (2.49) and (2.50) become Kronecker deltas δ_{ij}, and thus the "Hypothesis mixing" of Figure 2.6 simply becomes an identity operation (compare to Figure 2.2). As anticipated, the result is equivalence between the IMM algorithm and the MMAE algorithm (without *ad hoc* lower bounding of probabilities or restarting of divergent filters in the MMAE). Thus, the basic MMAE algorithm as developed in the opening of Section 2.2.4 is, in essence, a special case of an IMM in which the Markov matrix is an identity.

This research will concentrate on comparing the performance of an MMAE with *ad hoc* modifications to that of the basic IMM algorithm. In practice, a significant difficulty in IMM design (relative to MMAE design) is the specification of the entries of the Markov transition matrix — this in itself can be rather *ad hoc* and more difficult to accomplish well than specifying a single lower bound for probability values. Chapter 3 will discuss in more detail the specific techniques used to make these algorithms function, using a variety of filter types with different state dimensionalities.

2.3 Data Association and Hypothesis Reduction

The general problems of data association and hypothesis reduction have been well-researched. A recent technical paper by Williams and Maybeck [31] describes fairly succinctly the fundamental attributes of a target tracking problem in the presence of measurement clutter. It addresses data association methods for matching targets with measurements, Gaussian mixtures as representations of

potential target/measurement matches, and mixture reduction algorithms to facilitate computation. Furthermore, it introduces a new mixture reduction algorithm developed by Williams [30] based on minimizing the Integral Square Error cost measure. The ISE algorithm will form the basis for this research, so much of the technical paper's content appears below, in an adapted form.

In a typical target tracking scenario, each scan of a sensor produces a group of measurements, one or more of which may have been produced by the target(s) of interest and the rest produced by clutter. Clutter measurements may represent atmospheric effects, background objects, background targets of no interest, etc. Since the target tracking algorithm is attempting to track *only* targets of interest, there must be some provision for discerning which measurements are associated with real targets and which are associated with clutter. This process is known as *data association*, and for the purposes of this research, all problems will involve a single target of interest.

Generally speaking, it is not possible to make these measurement-target assignments in a single scan. Rather, the algorithm must collect hypothesized assignments across scans and perform tracking actions on each hypothesized association as if it was correct, until it is more clear which hypotheses were, indeed, incorrect from the beginning. Furthermore, limitations on computer memory and computation cycles mandate discarding the most unlikely hypotheses as time progresses. The process of optimizing the elimination of hypotheses is known as *hypothesis reduction*, and it takes place separate from the tracking portion of the algorithm.

2.3.1 Measurement Gating. Define the k-th scan to be a particular collection of measurements from a sensor at a given discrete time index k. Assume one or more targets are known to exist within this scan, and each target, i is predicted to produce a measurement at the location $\hat{\boldsymbol{z}}_i(k|k-1)$. Measurement gating reduces the number of measurements within the scan region that must be considered for data association. A typical gating relationship for deciding whether the measurement j should be included in the association is

$$[\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_i(k|k-1)]^{\mathrm{T}} \mathbf{A}_i(k)^{-1} [\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_i(k|k-1)] \leq \gamma \tag{2.54}$$

where $\mathbf{A}_i$ is the covariance of the residual formed using the predicted measurement belonging to target i, and γ is a threshold calculated using the desired probability that the correct measurement is within the gate (P_g) using χ^2 tables [4]. The j-th measurement is then said to be inside target i's association gate.

2.3.2 State Updates with Measurement Association Uncertainty. The problem considered is that of tracking a single target in clutter, although the technique proposed could be applied to

any problem in which the number of components in a Gaussian mixture is to be reduced. Denoting the target state at time index k as $\boldsymbol{x}(k)$, the measurement produced by the target at time k as $\boldsymbol{z}(k)$, and assuming a linear dynamics model with additive white Gaussian noise and a linear measurement model for the target-originated measurement, the true target state evolves according to the standard equation:

$$\boldsymbol{x}(k) = \boldsymbol{\Phi}(k, k-1)\boldsymbol{x}(k-1) + \mathbf{B_d}(k-1)\mathbf{u}(k-1) + \mathbf{w}(k-1) \quad (2.55)$$

$$\boldsymbol{z}(k) = \mathbf{H}(k)\boldsymbol{x}(k) + \mathbf{v}(k) \quad (2.56)$$

where $\mathbf{u}(k-1)$ is a known control input vector, $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are independent, zero-mean, white Gaussian noise processes with covariance $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ respectively, and $\boldsymbol{\Phi}(k, k-1)$, $\mathbf{B_d}(k-1)$, $\mathbf{H}(k)$, $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ are known matrices. If the initial PDF of target state is Gaussian, then the standard Kalman filter propagation and update equations [18] can be used to propagate the PDF from time-index to time-index, and to incorporate new measurements. Mild nonlinearities can also be admitted using the approximation provided by the extended Kalman filter [17].

We now switch to the case of interest in which, rather than receiving a single target-originated measurement $\boldsymbol{z}(k)$ at each time step, we receive a set of measurements $\boldsymbol{Z}_k$, which may or may not contain a target-originated measurement, alongside zero or more clutter measurements (false alarms). Denoting the measurements received up to time $(k-1)$ as $\boldsymbol{Z}^{k-1} = \{\boldsymbol{Z}_1, ..., \boldsymbol{Z}_{k-1}\}$, the a priori PDF (i.e., prior to introduction of the set of measurements at time k) of target state at time k can be written using a total probability expansion over the $N_h(k-1)$ association history hypotheses $\{\boldsymbol{\Psi}_u(k-1)\}_{u=1}^{N_h(k-1)}$ from the previous processing cycle:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}\} = \sum_{u=1}^{N_h(k-1)} f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}, \boldsymbol{\Psi}_u(k-1)\} P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\} \quad (2.57)$$

where $\{\boldsymbol{\Psi}_u(k-1)\}_{u=1}^{N_h(k-1)}$ represent the set of $N_h(k-1)$ association history hypotheses arising from measurement sets up to that received at time $(k-1)$, $P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}$ is the probability of the u-th association history hypothesis, and $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}, \boldsymbol{\Psi}_u(k-1)\}$ is the target state PDF conditioned on the u-th association history hypothesis. If the PDF at the previous time index conditioned on an association history hypothesis $f\{\boldsymbol{x}(k-1)|\boldsymbol{Z}^{k-1}, \boldsymbol{\Psi}_u(k-1)\}$ is Gaussian and the linear additive dynamics model of Eq. (2.56) applies, then the standard Kalman filter propagation equation can be used to calculate $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}, \boldsymbol{\Psi}_u(k-1)\}$, which is also Gaussian.

In order to introduce the set of $N_z(k)$ measurements at time k ($\boldsymbol{Z}_k$), a disjoint partitioning of the probability space $\{\psi_0(k), ..., \psi_{N_z}(k)\}$ is introduced, in which $\psi_0(k)$ denotes the event proposing

that the target was not detected at time k (and hence that all $N_z(k)$ measurements in $\boldsymbol{Z}_k$ are the result of clutter), and $\psi_i(k)$, $i = \{1, ..., N_z(k)\}$ denotes the event proposing that measurement i originated from the target (and the other $[N_z(k) - 1]$ measurements originated from clutter). The PDF of target state at time k conditioned on the new set of measurements can the be evaluated through the double-expansion over previous association history hypotheses $\{\boldsymbol{\Psi}_u(k-1)\}_{u=1}^{N_h(k-1)}$ and new association events $\{\psi_i\}_{i=0}^{N_z(k)}$:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{u=1}^{N_h(k-1)} \sum_{i=0}^{N_z(k)} f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \psi_i(k), \boldsymbol{\Psi}_u(k-1)\} P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} \tag{2.58}$$

where the upper limit of the first summation, $N_h(k-1)$, represents the number of components from the previous sample period assuming *no hypothesis reduction has taken place.* This upper limit will become $N_r(k-1)$ in later developments to denote the number of components in existence after hypothesis reduction.

Due to the conditioning on the association event $\psi_i(k)$ (which prescribes which measurement in $\boldsymbol{Z}_k$, if any, was target-originated), the Gaussian PDF $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \psi_i(k), \boldsymbol{\Psi}_u(k-1)\}$ can be calculated from the Gaussian PDF $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}, \boldsymbol{\Psi}_u(k-1)\}$ using the standard Kalman filter update equations, and again the posterior will remain Gaussian. Therefore, assuming that the prior PDF is a Gaussian mixture, the posterior PDF $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\}$ will also be a Gaussian mixture. The double summation of Eq. (2.58) can be combined for convenience into a single summation over an equivalent set of indices. Defining a new set of association history hypotheses $\{\boldsymbol{\Psi}_{u'}(k)\}_{u'=1}^{N_h(k)}$ where $N_h(k) = [N_h(k-1)][(N_z(k)+1)]$, Eq. (2.58) can be written as:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{u'=1}^{N_h(k)} f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \boldsymbol{\Psi}_{u'}(k)\} P\{\boldsymbol{\Psi}_{u'}(k)|\boldsymbol{Z}^k\} \tag{2.59}$$

where:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \boldsymbol{\Psi}_{u'}(k)\} = f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \psi_i(k), \boldsymbol{\Psi}_u(k-1)\} \tag{2.60}$$

$$P\{\boldsymbol{\Psi}_{u'}(k)|\boldsymbol{Z}^k\} = P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} \tag{2.61}$$

The association history event probabilities $P\{\boldsymbol{\Psi}_{u'}(k)|\boldsymbol{Z}^k\}$ are commonly referred to as probability weights, and are calculated using Bayes rule: (omitting unnecessary conditionings) as:

$$P\{\boldsymbol{\Psi}_{u'}(k)|\boldsymbol{Z}^k\} = P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} = P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}, \boldsymbol{Z}_k, N_z(k)\}$$
$$= \frac{P\{\boldsymbol{Z}_k|\psi_i(k), \boldsymbol{\Psi}_u(k-1), \boldsymbol{Z}^{k-1}, N_z(k)\}P\{\psi_i(k)|N_z(k)\}P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}}{P\{\boldsymbol{Z}_k|\boldsymbol{Z}^{k-1}, N_z(k)\}} \tag{2.62}$$

where $P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}$ is the association history hypothesis probability from the previous processing cycle, and the remaining terms are evaluated according to the standard model [4].

The challenge of tracking in the presence of clutter is clear from the expression $N_h(k) = [N_h(k-1)][(N_z(k)+1)]$: the number of components in the Gaussian mixture of Eq. (2.59) grows exponentially as new sets of measurements are received, and so it is necessary to perform some form of approximation to limit the growth.

2.3.3 Hypothesis Reduction. The previous section clearly illustrated the drastic effect on computation imposed by the spawning of hypotheses at each measurement update. Much effort has been devoted to developing algorithms for deciding which components in a Gaussian mixture (i.e., which association histories) can be eliminated at each sample period. Each technique must rely on a particular standard or metric for component elimination or merging, and it is those standards that essentially define the algorithm's performance. The goal is to create a lower order mixture that approximates a higher order mixture with sufficient fidelity not to affect the current target track adversely.

2.3.4 ISE Cost Function. We define the original PDF of target state, containing $N_h(k)$ joint hypotheses as to the possible location of the target, as $f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\}$ where $\boldsymbol{\Omega}_{N_h}(k)$ represents the parameters of the $N_h(k)$ hypotheses derived from the measurements up to the current sample period (probability weights, means and covariances). Our goal is thus to reduce these $N_h(k)$ hypotheses to a simplified representation, containing $N_r(k)$ hypotheses (the subscript r denoting reduced), resulting in the simplified PDF $f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\}$ where $\bar{\boldsymbol{\Omega}}_{N_r}(k)$ represents the reduced set of parameters, containing, as closely as possible, the same information as the original set $\boldsymbol{\Omega}_{N_h}(k)$.

A distance measure commonly used in nonparametric statistics is Integral Square Error (ISE): [12]

$$J_s = \int \left(f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\} - f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\}\right)^2 d\boldsymbol{x}(k) \tag{2.63}$$

The measure is very similar to the Kolmogorov variational distance used by Alspach [1], except that the even nonlinearity provided by the absolute value sign (which forces positive differences and negative differences to be handled identically) of the Kolmogorov variational distance measure is replaced with a square function. The nonlinearity could be replaced with any even integer power, where higher powers will tend to treat areas of larger error with increasingly higher weight than those of lower error. In the limit, as the power approaches infinity (in an even sense), the cost function will apply all priority to the largest error point, tending to *minimize* the *maximum point-wise error* committed by the approximation. The implication of this to the ISE measure is that it will behave similarly to the Kolmogorov variational distance, but comparatively higher weight will be applied to areas of larger error (e.g., components with smaller variance, and higher peaks), while comparatively lower weight will be applied to areas with lower error (e.g., components with larger variance, and flatter, broader peaks). This behavior is well-known in the nonparametric statistics community, and has been documented to provide greater immunity against outlying points than the Kullback-Leibler distance [27], another alternative to ISE.

Having considered the cost function options, the ISE distance is particularly attractive due to its tractability: it can be evaluated in closed from without approximation or numerical integration. Expanding the ISE distance measure equation yields the following terms:

$$J_s = \int f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\}^2 - 2f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\}f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\} + f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\}^2 d\boldsymbol{x}(k) \quad (2.64)$$

The three terms of Eq. (2.64) each have their own interpretation. The first represents the *self-likeness* of the original PDF — this term will be larger if the PDF is more concentrated in space, and smaller if the PDF is more spread out. The second represents the *cross-likeness* of the original PDF to the new PDF. This term is critical to the function as it directly measures the volume of probability mass that the two functions have in common (although it is in square units rather than the units typically associated with probability mass measure). The final term is the self-likeness of the reduced PDF, possessing similar characteristics to the other self-likeness term. The cross-likeness term serves to balance the two self-likeness terms, cancelling the overall cost function value to zero if the two functions are identical, and increasing the overall cost function value as the difference between the functions increases. Defining these three components as:

$$\begin{aligned} J_{hr} &= \int f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\}f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\}\mathrm{d}\boldsymbol{x}(k) \\ J_{rr} &= \int f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\}^2\mathrm{d}\boldsymbol{x}(k) \\ J_{hh} &= \int f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\}^2\mathrm{d}\boldsymbol{x}(k) \end{aligned} \quad (2.65)$$

we can then write Eq. (2.64) as:

$$J_S = J_{hh} - 2J_{hr} + J_{rr} \tag{2.66}$$

In the problem of interest, the two PDFs are both Gaussian mixtures, which can be expanded as:

$$\begin{aligned} f\{\boldsymbol{x}(k)|\boldsymbol{\Omega}_{N_h}(k)\} &= \sum_{i=1}^{N_h(k)} p_i \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i\} \\ f\{\boldsymbol{x}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\} &= \sum_{i=1}^{N_r(k)} \bar{p}_i \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_i, \bar{\mathbf{P}}_i\} \end{aligned} \tag{2.67}$$

where $\{p_i, \boldsymbol{\mu}_i, \mathbf{P}_i\}$ for $i = 1, 2, \ldots, N_h(k)$ are the weights, means and covariances of the Gaussian functions composing the mixture for original PDF, and $\{\bar{p}_i, \bar{\boldsymbol{\mu}}_i, \bar{\mathbf{P}}_i\}$ for $i = 1, 2, \ldots, N_r(k)$ are the same parameters of the reduced PDF. Substituting these expressions into Eq. (2.65):

$$\begin{aligned} J_{hr} &= \int \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_r(k)} p_i \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i\} \bar{p}_j \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_j\} \mathrm{d}\boldsymbol{x}(k) \\ J_{rr} &= \int \sum_{i=1}^{N_r(k)} \sum_{j=1}^{N_r(k)} \bar{p}_i \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_i, \bar{\mathbf{P}}_i\} \bar{p}_j \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_j\} \mathrm{d}\boldsymbol{x}(k) \\ J_{hh} &= \int \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_h(k)} p_i \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i\} p_j \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_j, \mathbf{P}_j\} \mathrm{d}\boldsymbol{x}(k) \end{aligned} \tag{2.68}$$

The product of two Gaussian PDFs, which forms the basic building block of Eq. (2.68), can be simplified to the following form:

$$\mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_1, \mathbf{P}_1\} \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_2, \mathbf{P}_2\} = \alpha \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_3, \mathbf{P}_3\} \tag{2.69}$$

where α, $\boldsymbol{\mu}_3$ and $\mathbf{P}_3$ are given by:

$$\begin{aligned} \alpha &= \mathcal{N}\{\boldsymbol{\mu}_1; \boldsymbol{\mu}_2, \mathbf{P}_1 + \mathbf{P}_2\} \\ \mathbf{P}_3 &= ({\mathbf{P}_1}^{-1} + {\mathbf{P}_2}^{-1})^{-1} \\ \boldsymbol{\mu}_3 &= \mathbf{P}_3({\mathbf{P}_1}^{-1}\boldsymbol{\mu}_1 + {\mathbf{P}_2}^{-1}\boldsymbol{\mu}_2) \end{aligned} \tag{2.70}$$

Substituting this simplification into the expressions of Eq. (2.68), each of the integral operations is the integral over the entire space of a Gaussian, evaluating to unity, leaving only the volume scaling

factors (α in Eq. (2.69)):

$$
\begin{aligned}
J_{hr} &= \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_r(k)} p_i \bar{p}_j \mathcal{N}\{\boldsymbol{\mu}_i; \bar{\boldsymbol{\mu}}_j, \mathbf{P}_i + \bar{\mathbf{P}}_j\} \int \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_a, \mathbf{P}_a\} \mathrm{d}\boldsymbol{x}(k) \\
&= \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_r(k)} p_i \bar{p}_j \mathcal{N}\{\boldsymbol{\mu}_i; \bar{\boldsymbol{\mu}}_j, \mathbf{P}_i + \bar{\mathbf{P}}_j\} \\
J_{rr} &= \sum_{i=1}^{N_r(k)} \sum_{j=1}^{N_r(k)} \bar{p}_i \bar{p}_j \mathcal{N}\{\bar{\boldsymbol{\mu}}_i; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_i + \bar{\mathbf{P}}_j\} \int \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_b, \mathbf{P}_b\} \mathrm{d}\boldsymbol{x}(k) \\
&= \sum_{i=1}^{N_r(k)} \sum_{j=1}^{N_r(k)} \bar{p}_i \bar{p}_j \mathcal{N}\{\bar{\boldsymbol{\mu}}_i; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_i + \bar{\mathbf{P}}_j\} \\
J_{hh} &= \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_h(k)} p_i p_j \mathcal{N}\{\boldsymbol{\mu}_i; \boldsymbol{\mu}_j, \mathbf{P}_i + \mathbf{P}_j\} \int \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_c, \mathbf{P}_c\} \mathrm{d}\boldsymbol{x}(k) \\
&= \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_h(k)} p_i p_j \mathcal{N}\{\boldsymbol{\mu}_i; \boldsymbol{\mu}_j, \mathbf{P}_i + \mathbf{P}_j\}
\end{aligned}
\tag{2.71}
$$

Interpreting Eq. (2.66) and Eq. (2.71), the cost function consists of the sum of similarity measures of all pairs of two components from the original mixture, plus similarity measures of all pairs of two components from the reduced mixture, balanced by the sum of similarity measures of all pairs of one component from the original mixture and one component from the reduced mixture.

Three benefits of ISE were mentioned by Williams. It has a closed form solution resulting in a sum of multivariate Gaussians with one term for each pairing of components in the original and reduced mixtures, unlike any of the previous cost functions [30] that had been used to address mixture reduction. The cost evaluation is continuously differentiable, allowing standard gradient techniques to be used for function optimization. Finally, expressions for the gradient can be written in vector-matrix notation, thereby simplifying further the use of gradient-based iterative optimization. The last two benefits are less important than the first, due to the fact that Williams' final algorithm did not employ a gradient search.

2.3.5 ISE Reduction Algorithm. The initial intent of Williams' research was to perform the hypothesis reduction by applying a gradient-based iterative optimization algorithm to find the parameters of the set of reduced mixture components which provide the best fit to the full mixture. The derivatives of the ISE cost function can also be evaluated in closed form using standard vector-matrix notation [30], and so gradient algorithms can be easily applied. However, the cost function describing the fit of a reduced complexity Gaussian mixture to a Gaussian mixture of higher order is an extremely complicated multi-modal function with many local minima, so the local view provided

by gradient-based methods will not provide any promise of convergence to a global minimum, unless the initialization point happens to be close to the global minimum.

One way of restricting the optimization space is to consider only those solutions corresponding to a sequence of merging and pruning operations. Solutions to this problem correspond to assignments, in which each component in the input mixture is assigned to zero or one component in the output mixture (zero if the input component is pruned), and each component in the output mixture is assigned one or more components in the input mixture. By enumerating all such assignments, and evaluating their cost, a large region of the solution space is explored, and the lowest cost result is likely to be close to the global minimum.

The obvious problem with this approach is that the number of possible assignments is combinatorially large for cases in which the number of input components is large. To reduce the computational complexity to a manageable level, a greedy assignment solution was adopted, which commences from the original PDF, and merges and prunes components until the number of components has been reduced to the desired level. At each stage of the algorithm, the cost of all possible merging and pruning actions is evaluated, and the action with the lowest cost is taken. A functional block diagram of the ISE reduction process appears in Figure 2.7.

Although the resulting algorithm is similar in concept to the joining algorithm proposed by Salmond, [23–26] it has a significant advantage in that the ISE cost function considers the *entire mixture* (i.e., the entire PDF) when making merging and pruning decisions, rather than individual pairs of mixture components in isolation.

When two components are merged, the parameters of the merged component are calculated such that the mean and covariance of the overall mixture remains unchanged:

$$\text{Weight}: \quad p_c = p_1 + p_2 \tag{2.72}$$

$$\text{Mean}: \quad \mu_c = \frac{1}{p_1 + p_2}\left\{p_1\hat{\mathbf{x}}_1 + p_2\hat{\mathbf{x}}_2\right\} \tag{2.73}$$

$$\text{Covariance}: \quad \mathbf{P}_c = \frac{1}{p_1 + p_2}\left\{p_1\mathbf{P}_1 + p_2\mathbf{P}_2 + \frac{p_1 p_2}{p_1 + p_2}[\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2][\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2]^{\mathrm{T}}\right\} \tag{2.74}$$

When components are deleted, the probability weights of the remaining components are not increased. This choice was made on the basis of the geometrical observation that renormalizing weights will increase the error value in regions of the PDF which are unaffected by the deletion. Logically it would seem that the only error increase incurred by deleting a component should come from the difference created in the region previously occupied by the component, and that differences

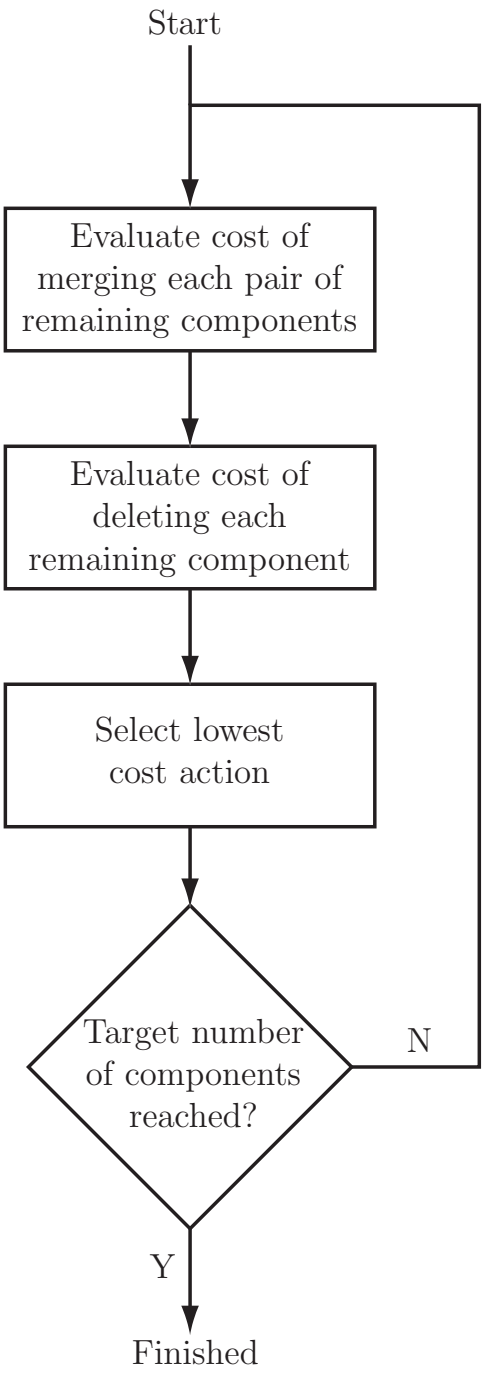

Figure 2.7: ISE algorithm functional block diagram

elsewhere in the PDF should not come into consideration. The choice also provides a substantial computational saving, discussed in the following section.

2.3.6 ISE-Based Algorithm Performance. In the initial stages of research [30], an extensive evaluation of performance of the ISE-based technique was accomplished via MATLAB® simulations. Track life performance of the technique was compared to that of an n-best pruning technique and to Salmond's joining and clustering filters (the best algorithms known heretofore for handling a single target in dense clutter), utilizing between 1 and 40 mixture components. The results showed that the performance of the ISE-based method is significantly better than those of the other methods, when the comparison is performed using the same number of components in the output mixture for each method. Fig. 2.8 displays the results of this study.

Average track life was one of the major metrics used to compare algorithm performance in [23–26]. This figure clearly reveals the remarkable performance of the ISE-based algorithm (denoted the "Integrated Square Difference Initialization" or "ISD Initialization" algorithm at the time of these early results) using a large number of mixture components. The average track life is seen to be significantly greater than that of the algorithms which had previously been considered to provide best performance in this scenario. Moreover, the other algorithms exhibit an average track life that plateaus as the number of mixture components is increased, indicating that additional computer resources would provide little performance benefit. In contrast, the exponential increase of track life shown by the ISE-based technique indicates that the tracking performance is limited only by the availability of computational resources.

Because the computational complexity of the various algorithms differ substantially, it is not necessarily meaningful to compare those algorithms only on the basis of using the same number of mixture components. The computational complexity of the algorithms is difficult to establish analytically, as it is affected by a wide range of interacting phenomena, such as the covariance of the components maintained (if covariances increase, then more measurements enter the association gate, greatly increasing the work associated with simplifying the representation). Furthermore, since each of the algorithms can be vectorized to a different extent, simulations running in MATLAB® do not provide a consistent view of the computational complexity of the methods.

In order to evaluate the performance of the ISE-based algorithm against previous methods on the basis of computational complexity, a C++ simulation was developed [31], incorporating reduction methods including the ISE-based method, the Salmond joining filter, a simple pruning technique, the n-scan filter [28], and the Mixture Kalman Filter (MKF) [8] — a variant of the popular particle filter. Numerical Recipes [21] routines were utilized for random number generation and basic matrix operations. Implementation decisions for the simulation included:

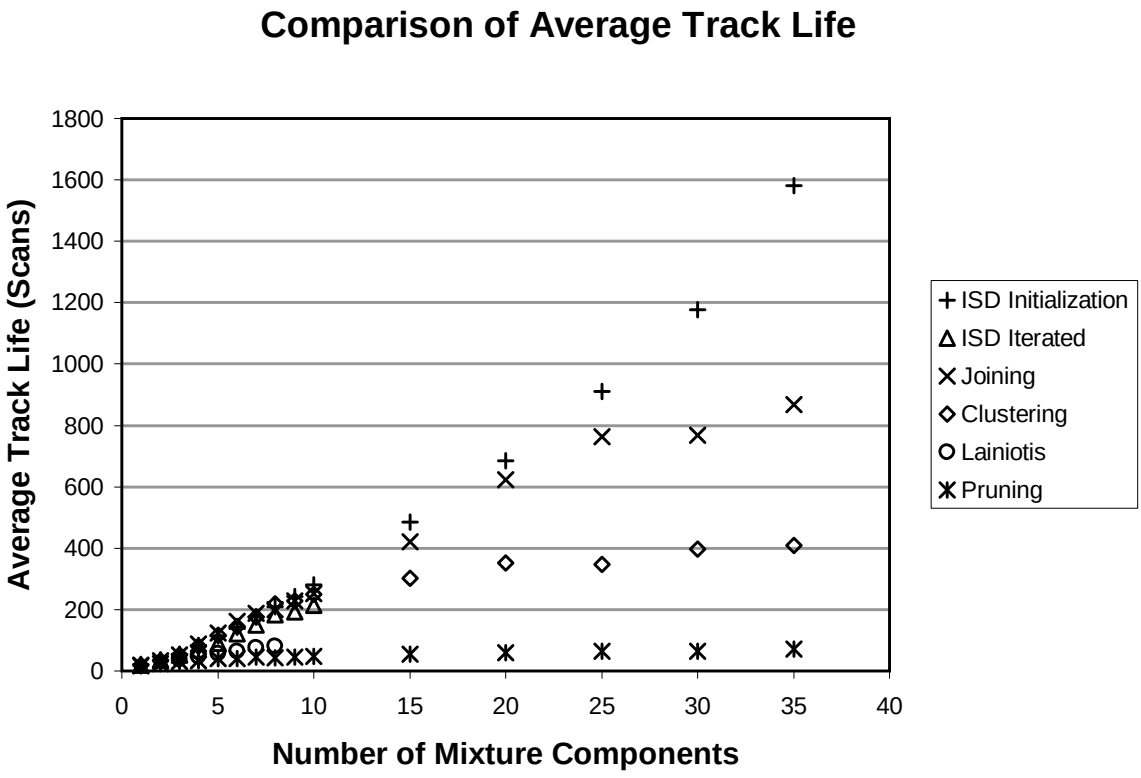

Figure 2.8: Average track life: ISE algorithm performance versus alternative methods with a probability of target detection equal to 1.0

- Individual association gates were formed for each association history hypothesis. This decision was made to allow hypotheses to explore as large a region of the measurement space as the algorithm made possible, and to control the number of hypotheses generated.
- For all methods, the maximum number of hypotheses generated by any single parent association history hypothesis was limited to the 50 measurements closest to the predicted measurement. This limit was imposed to prevent the massive computational increase produced when component covariances become progressively larger, progressively allowing more measurements to enter the association gate for the hypothesis.
- To limit the growth of extremely low-probability hypotheses, the n-scan memory filter was modified to discard components with weights less than 10^{-6}.
- The MKF implementation used the systematic resampling with replacement method presented as Algorithm 2 in Arulampalam, et. al. [2]. When the same association history hypothesis was drawn several times, a single output component was created, with a weight corresponding to the number of times the hypothesis was drawn.
- The pruning algorithm discards the set of hypotheses with the smallest weight such that the renormalized weight of the smallest remaining component is greater than our equal to a given threshold. This implementation was chosen as it is adaptive (it keeps a larger set of components when the probability mass is split evenly over many hypotheses, and a smaller set of hypotheses when the probability mass is dominated by fewer hypotheses), and it limits the number of components retained at any processing interval to the reciprocal of the threshold.
- The implementation of the Salmond joining filter used the recommended threshold values given in [23]. The performance of the technique could potentially be improved by tuning the thresholds to the test scenario.

The test scenario used [31] was an adapted version of the dense clutter single-target tracking simulation presented by Salmond [23]. The target state evolves according to the following constant-

velocity model:

$$\begin{aligned}
\boldsymbol{x}(k) &= \begin{bmatrix} p_x(k) \\ v_x(k) \\ p_y(k) \\ v_y(k) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(k-1) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \boldsymbol{w}(k-1) \\
\boldsymbol{z}(k) &= \begin{bmatrix} z_x(k) \\ z_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}(k) + \boldsymbol{v}(k)
\end{aligned} \tag{2.75}$$

where T is the time between measurement instants $(k-1)$ and k, and $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ are two independent zero-mean white Gaussian noise processes such that:

$$\begin{aligned}
E\{\boldsymbol{w}(k)\boldsymbol{w}(k)^{\mathrm{T}}\} &= \mathbf{Q} = q\mathbf{I} \\
E\{\boldsymbol{v}(k)\boldsymbol{v}(k)^{\mathrm{T}}\} &= \mathbf{R} = r\mathbf{I}
\end{aligned} \tag{2.76}$$

The system is provided with noise-corrupted measurements of the target position (x and y coordinates) through a linear measurement model; the system could be extended to polar measurements (i.e., range and angle) using the extended Kalman filter [17].

The parameters T, q and r were all normalized to unity, the clutter density λ was set to 0.012, and the probability of detection (P_d) was set to 0.8 and 0.9 for two sets of simulations (note that P_d was assumed to be 1.0 for the results in Fig. 2.8). The gate size was set such that the probability of the target-originated measurement being in the association gate was $P_g = 0.99$. The target was initially located at the origin with a velocity of 10 units/sec in each coordinate direction. Clutter was generated according to a Poisson distribution. The region populated was a square, centered on the actual target location, with side $2000\, r^{1/2}$. This value was chosen to be large such that hypotheses could be deceived by clutter measurements for several processing cycles without leaving the populated region. The expected number of false targets in each processing cycle for this configuration is 48,000. As mentioned above, gating was performed independently for each hypothesis. While an efficient two-stage gating algorithm [30] was utilized (incorporating a coarse rectangular gate, followed by an elliptical gate applied only to measurements passing the rectangular gate), the large number of measurements, coupled with the per-hypothesis gating, adds a substantial linear component to the computational cost of considering additional hypotheses, and the results should be qualified accordingly.

The criterion for loss-of-track was based on a comparison between the true target location and the location of each hypothesis, weighted according to the covariance matrix of the "aided"

Algorithm	Parameter								
Pruning	weight threshold	0.1	0.01	0.001	0.0005				
n**-Scan**	number of scans	1	2	3					
MKF	max # output comp	100	200	300	400	500	600	700	800
Joining	max # output comp	30	40	50	60	70			
ISE	# output comp	30	40	50	60				

Table 2.1: Parameter values used for simulations; for each, 200 Monte Carlo simulations were conducted.

Kalman filter (i.e., the Kalman filter assuming that the index of the target-originated measurement is known), $[\boldsymbol{x} - \hat{\boldsymbol{x}}_i]^{\mathrm{T}} \mathbf{P}_a^{-1} [\boldsymbol{x} - \hat{\boldsymbol{x}}_i]$, where $\boldsymbol{x}$ is the true target state, $\hat{\boldsymbol{x}}_i$ is the mean estimate under hypothesis i, and $\mathbf{P}_a$ is the covariance of the aided Kalman filter. Track-loss was declared when this statistic was greater than 10 (corresponding to a $10^{1/2}$-sigma deviation from the mean) for all association hypotheses maintained by the filter for five consecutive time steps. The criterion was based on all hypotheses rather than the combined estimate (i.e., the weighted average of hypothesis means) to allow the deferred decision-making ability of the multiple hypothesis formulation to take place: one anticipates that the combined estimate will diverge from the correct location for small periods of time before subsequent measurements clarify the association uncertainty, increasing the weight of the correct hypothesis, and returning the combined estimate to the vicinity of the true target. In order to monitor the behavior of the combined estimate, an additional error criterion was introduced, which measures the proportion of the time (before track loss is declared) in which the combined estimate is in the vicinity of the true target state according to the same metric and threshold used for the individual hypothesis comparisons in the loss-of-track criterion.

For each algorithm and each parameter value, 200 Monte Carlo simulations were conducted, each of which was allowed to run until loss-of-track was declared. Simulations were run on a 2.6 GHz Intel® Xeon® processor. The parameter points run for each algorithm are summarized in Table 2.1.

The simulation results are presented in Figure 2.9. The plots in (a) and (d) echo the findings of the earlier comparison (see Fig. 2.8) based on the number of output components (for different detection probabilities — left-hand plots in Fig. 2.9 correspond to $P_d = 0.8$ while right-hand plots correspond to $P_d = 0.9$). This reiterates the fact that the ISE-based technique performs substantially better than the other algorithms using the same number of components in the Gaussian mixture. The plots in (b) and (e) qualify these results, showing that, for the same level of computational complexity (i.e., mean computation time per time step), the MKF and the n-scan memory filter

can exceed the performance of the ISE-based technique. The plots in (c) and (f) compare the error of the combined estimates, rather than the minimum error for any hypothesis, indicating that the combined estimate calculated with the ISE-based method tends to remain close to the true target state for a larger portion of the time for points with similar mean track life.

2.4 Summary

Having identified two proven multiple model state estimation algorithms for handling changing trajectory properties, as well as an efficient and high-fidelity mixture reduction algorithm for addressing clutter, this research will integrate them to perform single-target tracking in the presence of clutter with the onset of target maneuvers. Performance will be evaluated based on blended state estimate error, elemental filter probability flows, overall track length (i.e., the time until track loss), and computational loading (using various criteria to be described in Chapter 3).

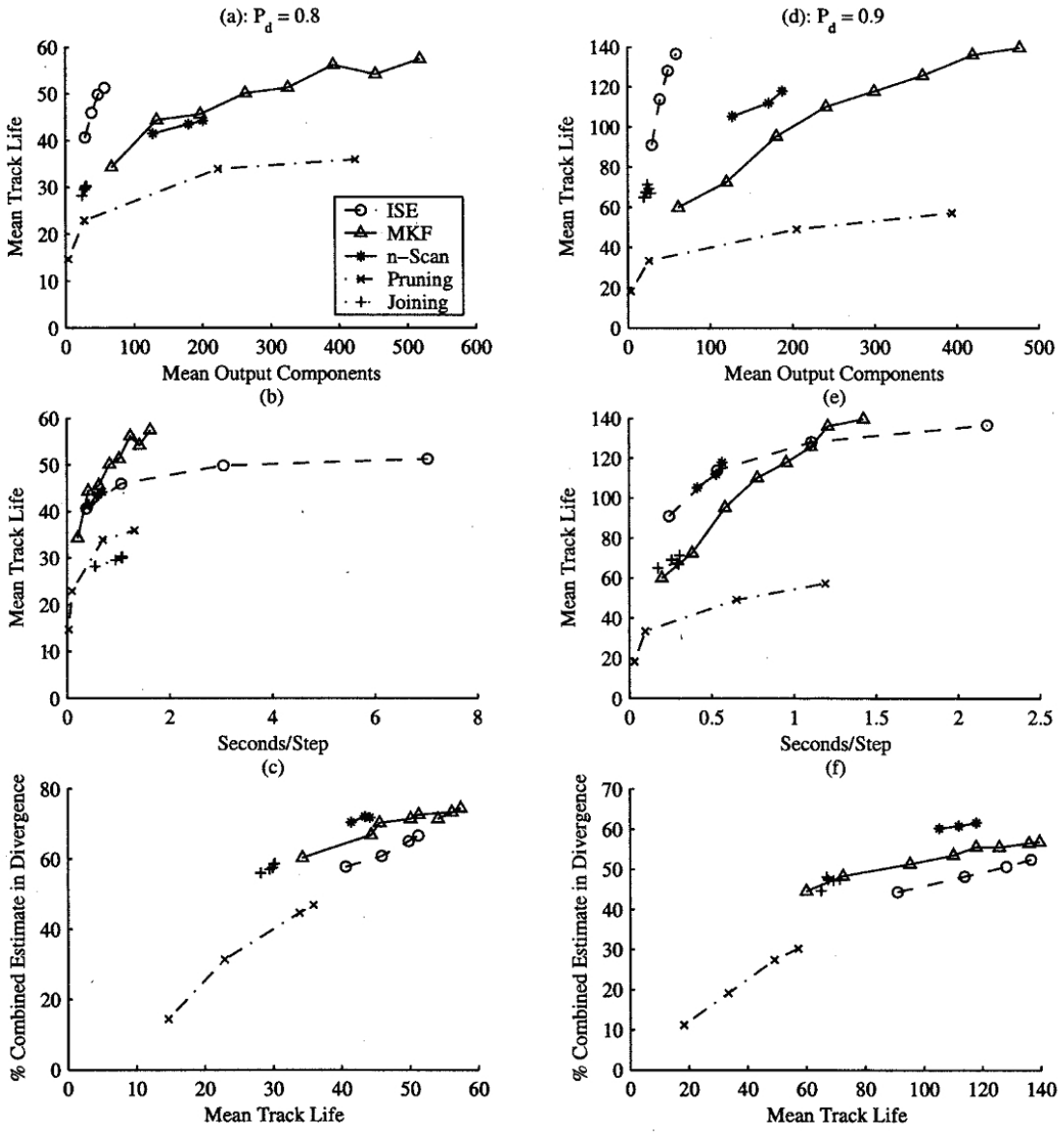

Figure 2.9: Comparison of mean track life vs. mean number of output components [(a),(d)], mean track life vs. mean computation time per time step [(b),(e)], and percentage of time steps in which combined estimate is diverged prior to loss-of-track vs. mean track life [(c),(f)]. Left-hand plots [(a),(b),(c)] correspond to $P_d = 0.8$, while right-hand plots [(d),(e),(f)] correspond to $P_d = 0.9$

III. Simulation Development and Analysis

3.1 Introduction

Chapter 2 made the case for using multiple model estimation algorithms for tracking targets with uncertain maneuver characteristics. Sections 2.2.4 and 2.2.5 discussed both switching-parameter and non-switching-parameter multiple model estimation algorithms applied to situations in which there is no measurement association uncertainty. The multiple-model adaptive estimator (MMAE) and the interacting mixed model (IMM) algorithms were seen as superior to a single filter estimator in terms of target tracking performance, especially state estimate rms error, when target dynamics are not known exactly. Both of these algorithms were seen to be more computationally feasible than either a full-order Bayesian estimator or a Second-Order Generalized Pseudo-Bayesian (GPB2) estimator [3], yet their tracking performance is on-par with these higher-order methods. The IMM is specifically intended to have the same performance characteristics as a GPB2 estimator, while only requiring the computational load of the First-Order Generalized Pseudo-Bayesian (GPB1) estimator [3]. Although the MMAE makes no claims about its performance compared to GPB2, it has shown excellent performance in a variety of applications. Therefore, for the purposes of this research, MMAE and IMM strike the right balance of target track fidelity and computational loading.

Section 2.3 introduced the multiple hypothesis tracker (MHT) concept as a means of target tracking in the presence of measurement association uncertainty. Section 2.3.3 discussed the need for hypothesis reduction in an MHT to prevent an exponential growth in computational requirements at every measurement cycle, and justification was made in Section 2.3.4 for using the Integral-Square Error as a cost function for hypothesis reduction. It exhibits mathematical tractability not found in similar cost functions, and excels in producing reduced-order Gaussian mixtures from higher-order mixtures without sacrificing state estimate fidelity and without the resulting tracker suffering from consistent loss of track.

This research involved the development of essentially four separate algorithms. The first two algorithms, a baseline MMAE and IMM, were developed for use in cases without measurement association uncertainty. These two designs represent the best performance one could expect when using multiple model algorithms in a target tracking problem, and they will provide a basis of comparison by which to judge the relative performance of the MHT algorithms. The second pair of algorithms, a multiple hypothesis tracker MMAE and multiple hypothesis tracker IMM, share many of the core characteristics of these non-MHT algorithms, but measurement association uncertainty will degrade their performance relative to their non-MHT counterparts. Section 3.2 discusses the implementation of block partitioning in state and error covariance calculation, a technique applied to both the non-MHT and MHT algorithms.

The development of the MHT MMAE and IMM algorithms and the necessary extensions to the non-MHT algorithms of Chapter 2 will also be discussed in this chapter. Section 3.3 will discuss, in detail, the replacement of the standard Kalman filters with Williams hypothesis reduction filters. Section 3.3.2 will discuss the formation of "pseudo-states" from the Gaussian mixture that serve the same purpose as the m-th elemental filter output states, $\hat{\boldsymbol{x}}_m(k|k-1)$ and $\hat{\boldsymbol{x}}_m(k|k)$, in the non-MHT algorithms. It also discusses the use of "pseudo-residuals" as substitutes for the true residuals in the non-MHT algorithms (which are uncertain in an MHT). These pseudo-residuals will be formed by matching incoming measurements with predictions based on pre-existing hypotheses (thereby creating a set of individual hypothesis residuals) and generating a probability-weighted average of these individual residuals based on their associated hypothesis probabilities. Section 3.4 will discuss appropriate means of determining whether an MHT has experienced track loss.

Section 3.5 will discuss *ad hoc* design choices made in the configuration of the MMAE and IMM algorithms. These choices, in general, impact parameters built into the various algorithms and do not represent any modifications to the algorithms themselves. Section 3.5.1 will summarize the wide variety of truth model/filter model combinations used in this research. Some of these combinations are commonly used in target tracking research, while others might be considered more realistic (for certain scenarios) or more challenging than the commonly accepted combinations. Section 3.5.2 will discuss the choices of MMAE parameters, and Section 3.5.3 will discuss the philosophy behind the IMM Markov probability transition matrices used in this research. Finally, the remainder of Section 3.5 will cover various parameters inherent to the MHT algorithms' functionality and the measurement "environment" in which they operate.

3.2 Block Calculation of State Estimates and Covariances

Traditional multiple-model algorithms often assume that all elemental filters have dynamics models with the same number of state variables. However, there is considerable utility in allowing the use of differently-dimensioned elemental filters within the same estimator structure. This is particulary useful in the target tracking case, because the target tracking algorithm is attempting to match one of its *assumed* models to the *real* target's dynamics, which at any given time, is best represented using a dynamics model with an arbitrary number of states.

Elemental filters of different dimension present several problems in practice for both MMAE and IMM. First, the tracker's blended state estimate and error covariance is formed as a probability-weighted sum of the individual filter state estimates and error covariances. Since these are summations of vectors and matrices, however, there is an implication that each filter has the same number of states. One option is to allow the tracker to output a combined estimate including only the states

common to all filters. For example, a tracker with two four-state filters and one six-state filter would produce a blended estimate with only four states. Unfortunately, this tracker ignores the information about states five and six from the six-state filter – information that might be useful in the particular tracking application. Ideally, the tracker would compute a blended estimate and error covariance for all six states, but it would need to account for the fact that only one filter (and its associated probability weight) is providing the estimate for states five and six.

Consider the conceptual system below, using the example provided above for which it is desirable to keep an estimate of all states available within the complete system:

$$\begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ x_{acc}(k) \\ y_{acc}(k) \end{bmatrix}_B = \begin{bmatrix} p_1 \\ \hline 0 \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \\ \\ \end{bmatrix}_{EF_1} + \begin{bmatrix} p_2 \\ \hline 0 \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \\ \\ \end{bmatrix}_{EF_2} + \begin{bmatrix} p_3 \\ \hline p_3' \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \hline x_{acc}(k) \\ y_{acc}(k) \end{bmatrix}_{EF_3} \tag{3.1}$$

where subscript B implies the tracker blended estimate, and subscripts EF_m for $m = 1, 2, 3$ implies the m-th elemental filter estimate with associated filter probability weight p_m. The six-state filter is broken down into two probability weights, p_3 and p_3', such that states one through four are scaled by p_3 and states five and six are scaled by p_3'. The probability p_3 is normalized by the sum of all filter probabilities in the system, and p_3' is always equal to one (since filter three is the only filter representing states five and six).

This "blocking" of state calculations can be generally extended to an arbitrary number of filters with an arbitrary number of states. For instance, consider an extension of Eq. (3.1) in which a fourth elemental filter is added to the MMAE. Let this fourth filter be an 8-state filter that also estimates jerk as well as position, velocity, and acceleration. Thus:

$$\begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ x_{acc}(k) \\ y_{acc}(k) \\ x_{jerk} \\ y_{jerk} \end{bmatrix}_B = \begin{bmatrix} p_1 \\ \hline 0 \\ \hline 0 \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \\ \\ \end{bmatrix}_{EF_1} + \begin{bmatrix} p_2 \\ \hline 0 \\ \hline 0 \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \\ \\ \end{bmatrix}_{EF_2} +$$

$$\begin{bmatrix} p_3 \\ \hline p_3' \\ \hline 0 \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \hline x_{acc}(k) \\ y_{acc}(k) \end{bmatrix}_{EF_3} + \begin{bmatrix} p_4 \\ \hline p_4' \\ \hline p_4'' \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ y_{pos}(k) \\ x_{vel}(k) \\ y_{vel}(k) \\ \hline x_{acc}(k) \\ y_{acc}(k) \\ \hline x_{jerk} \\ y_{jerk} \end{bmatrix}_{EF_4} \tag{3.2}$$

Note that we presume the same state ordering is used for all elemental filters for this to be useful. The partition-sensitive probability values indicate a top element (associated with the upper partition of the state vector, namely, the first four states, estimated by all four elemental filters) as computed within the MMAE itself, such that

$$p_1 + p_2 + p_3 + p_4 = 1$$

The middle element of the partition-sensitive probability values is associated with the middle partition of the state vector (the two acceleration states, estimated only by elemental filters 3 and 4), and these have been re-normalized to sum to one:

$$p'_3 + p'_4 = 1$$

Explicitly, they are re-normalized values of p_3 and p_4, scaled by the sum $[p_3 + p_4]$:

$$\begin{aligned} p'_3 &= p_3/[p_3 + p_4] \\ p'_4 &= p_4/[p_3 + p_4] \end{aligned}$$

Finally, the bottom element of the partition-sensitive probability values is associated with the bottom partition of the state vector (the two jerk states, estimated only by elemental filter 4), re-normalized to sum to one:

$$p''_4 = 1$$

Clearly, a similar procedure is required for the associated covariance calculations. Continuing with the example above, the upper left 4-by-4 partition of the covariance (associated with the two position states and two velocity states) would be computed as in Eq. (2.32), using a summation over all four elemental filters involving the original probability values, p_1, p_2, p_3 and p_4. The next two rows and columns that form the remainder of the upper left 6-by-6 partition (now incorporating acceleration state terms as well) would be computed as a summation over elemental filters three and four only, this time using the re-normalized values p'_3 and p'_4. Finally, the last two rows and columns to form the entire 8-by-8 matrix (now including jerk state terms as well) would be derived entirely from elemental filter 4.

The IMM mixing cycle will also need to incorporate filters with different numbers of state variables, and while the technique described above was used to calculate the tracker blended estimate, it can easily be adapted to the IMM mixing computations. Each filter in the IMM receives a mixed estimate and error covariance at the end of every cycle formed as a probability-weighted summation of

the estimates and covariances of every filter in the system (see Eqs. (2.49) and (2.50)). The weighting coefficients used in the mixing are calculated as shown in Eq. (2.47) and conceptually, for every left-hand side mixed estimate $\hat{\boldsymbol{x}}^i(k|k)$ there is a *vector* of mixing probabilities $P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\}$ for $j = 1, \ldots, N_f$. This vector of *mixing* probabilities can be substituted for the vector of *modal* probabilities, $p_1, \ldots, p_{N_f}$ in the partition-sensitive blending technique with the only one modification: the actual number of partitions being used in the computation of the i-th mixed estimate and error covariance is determined by the number of partitions represented in the i-th filter. So, while the algorithm must be carried out for each *filter* in the system, not every instance of the algorithm will require all *partitions* in the system. The primary complication in the mixing technique is that the scaling of mixing probabilities is partition sensitive. So, in addition to each partition having a vector of modal probabilities (one probability for each filter in the partition), it also has a matrix of mixing probabilities (one row and column of probabilities for each filter in the partition). As was the case for the modal probabilities, these partition-sensitive mixing matrices are simply re-normalized versions of the actual mixing matrix calculated by the IMM algorithm. When using a unity Markov probability transition matrix, this mixing exactly replicates internal state and error covariance feedback of the MMAE, further validating the algorithm (see Chapter 2, Section 2.2.9 for further explanation of the similarities and differences between IMM and MMAE).

3.3 Williams Filter Integration with a Multiple-Model Structure

3.3.1 Points of Entry and Exit. The developments for multiple model algorithms in Chapter 2 use standard Kalman filters in each elemental filter, and a full-order MHT, in theory, applies the standard Kalman filter to every component within the Gaussian mixture. In effect, each cycle of a full-order MHT spawns a new set of Kalman filters, each conditioned upon a particular measurement association history. Chapter 2 explained in detail the alternatives to full-order MHT's, and all of them require a reduction of the number of hypotheses at each time step. To replace a multiple-model estimator's standard Kalman filters with hypothesis reduction filters (Williams filters, in the case of this research), points of entry into the filter and points of exit out of the filter need to be established.

For instance, the previous state estimate $\hat{\boldsymbol{x}}(k-1|k-1)$ and associated error covariance are inserted into the entry point for a Kalman filter propagation cycle, and the exit point produces the propagated state estimate $\hat{\boldsymbol{x}}(k|k-1)$ and its associated error covariance $\mathbf{P}(k|k-1)$. The anticipated residual covariance, $\mathbf{A}(k) = [\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}(k)^{\mathrm{T}} + \mathbf{R}(k)]$ can also be produced at this exit point. Similarly, the entry point of the update cycle requires $\hat{\boldsymbol{x}}(k|k-1)$, the incoming measurement $\boldsymbol{z}(k)$, and the covariance $\mathbf{P}(k|k-1)$. At the exit of the update cycle, the residual $\boldsymbol{r}(k) = [\boldsymbol{z}(k) - \mathbf{H}(k)\hat{\boldsymbol{x}}(k|k-1)]$,

the updated state estimate $\hat{\boldsymbol{x}}(k|k)$, and the associated error covariance $\mathbf{P}(k|k)$ are produced. If a Williams filter is to replace a Kalman filter as the form of elemental filter within an MMAE or IMM, then the vectors and matrices that play the corresponding roles at the entry and exit points of Williams filter propagations and updates must be identified. Recall, for example, that the residual $\boldsymbol{r}(k)$ and its covariance $\mathbf{A}(k)$ are essential to the probability computations within an MMAE or IMM.

In a Williams filter propagation cycle, the conditional state density for each pre-existing hypothesis, $f\{\boldsymbol{x}(k-1)|\boldsymbol{Z}^{k-1}, \Psi_u(k-1)\}$ for $u = 1, \ldots, N_r(k-1)$, is propagated forward in time to (k), where $N_r(k-1)$ represents the number of hypotheses existing after reduction in the previous cycle, $(k-1)$. Minimal modification of an MMAE or IMM algorithm is required to accommodate this type of propagation, and the exit point of the Williams propagation will produce both a filter *pseudo-state* $\hat{\boldsymbol{x}}(k|k-1)$ (to be discussed in more detail in Section 3.3.2) and associated filter *pseudo-error covariance*, and an array of Gaussian components $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}, \Psi_u(k-1)\}$ for $u = 1, \ldots, N_r(k-1)$.

After gating incoming measurements, the MHT tracker enters its update cycle. Each pre-existing hypothesis is associated with each gated measurement resulting in

$$N_h(k) = N_r(k-1)N_z(k) + N_r(k-1)$$

new hypotheses, where $N_h(k)$ is the number of hypotheses that will enter the Williams reduction cycle, and $N_z(k)$ is the number of incoming measurements that passed the appropriate measurement gate. The additional $N_r(k-1)$ term accounts for the missed detection cases which result in each pre-existing hypothesis being carried over to time step $(k-1)$ *without a measurement association and update*. The resulting components, $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \psi_i(k), \Psi_u(k-1)\}$ for $i = 0, \ldots, N_z(k)$ and $u = 1, \ldots, N_r(k-1)$, will be passed into the Williams filter reduction cycle to be reduced from $N_h(k)$ to $N_r(k)$ number of hypotheses. Like the exit point of the propagation cycle, the exit point of the update cycle after reduction produces both a filter pseudo-state $\hat{\boldsymbol{x}}(k|k)$ and pseudo-error covariance $\mathbf{P}(k|k)$ and an array of Gaussian components $f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k, \Psi_u(k)\}$ for $u = 1, \ldots, N_r(k)$.

Specifying these points of entry and exit have illuminated the locations in the algorithms where state estimates, $\hat{\boldsymbol{x}}(k|k-1)$ and $\hat{\boldsymbol{x}}(k|k)$, and their associated error covariances are passed. Consequently, the standard Kalman filter can be replaced by a "black box" filter that has similar entry and exit points and passes similar data. What happens within the black box is entirely independent of whether the filter is part of a multiple-model system or is stand-alone.

3.3.2 Pseudo-State Estimates, Error Covariances, and Residuals. In the standard Kalman filter, $\hat{\boldsymbol{x}}(k|k-1)$, $\hat{\boldsymbol{x}}(k|k)$, and their associated error covariances are obvious. In the MHT algorithm,

these estimates do not exist as outputs of a single filter, because the outputs of individual Kalman-like filters are conditioned upon a particular measurement association history. Furthermore, multiple-model algorithms require a residual term $[\boldsymbol{z}(k) - \mathbf{H}\hat{\boldsymbol{x}}(k|k-1)]$ and associated covariance to calculate modal probabilities. Again, a single residual does not exist in an MHT, because there are multiple measurements and multiple hypotheses conditioned on multiple prior measurement association histories. However, applying the "black box" approach outlined in the previous section, one can envision *pseudo-state estimates*, *pseudo-state error covariances*, *pseudo-residuals*, and *pseudo-residual error covariances*.

Consider the calculation of the blended estimate within a multiple-model tracker. As shown in Chapter 2, the blended estimate and error covariance are formed as the probability-weighted sum of all filter estimates and error covariances (see Eqs. (2.31) and (2.32)). Similarly, since the individual components in each hypothesis reduction elemental filter are Gaussians (under the assumptions made in Chapter 2) with assigned probability weights, a blended estimate could be formed using the results of *all* such Gaussian densities. In Chapter 2, the equation for the conditional pdf of the target state conditioned upon the measurement history through instant $(k-1)$ was given in Eq. (2.57) and repeated here for convenience:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}\} = \sum_{u=1}^{N_h(k-1)} f\{\boldsymbol{x}(k)|\boldsymbol{Z}^{k-1}, \boldsymbol{\Psi}_u(k-1)\} P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\} \tag{3.3}$$

where $\boldsymbol{Z}^{k-1}$ is the measurement history through time instant $(k-1)$, $N_h(k-1)$ is the number of association hypotheses retained in the Gaussian mixture as of time instant $(k-1)$, $\boldsymbol{\Psi}_u(k-1)$ is one such association history hypothesis for $u = 1, \ldots, N_h(k-1)$, and $P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}$ is the probability of the u-th association hypothesis. Note that each density is produced as the output of a single conventional Kalman filter, since the association history is assumed to be known.

From the expression in Eq. (3.3), the conditional mean of the m-th elemental filter's state $\hat{\boldsymbol{x}}_m(k|k-1)$ and its corresponding state error covariance $\mathbf{P}_m(k|k-1)$ can be seen to be computable as the analogous $N_h(k-1)$-term sums that are structurally identical to the standard MMAE state estimate and error covariance:

$$\hat{\boldsymbol{x}}_m(k|k-1) = \sum_{u=1}^{N_h(k-1)} \hat{\boldsymbol{x}}_{m,u}(k|k-1) \cdot P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}$$

$$\mathbf{P}_m(k|k-1) = \sum_{u=1}^{N_h(k-1)} \{\mathbf{P}_{m,u}(k|k-1)+ [\hat{\boldsymbol{x}}_{m,u}(k|k-1) - \hat{\boldsymbol{x}}_m(k|k-1)][\hat{\boldsymbol{x}}_{m,u}(k|k-1) - \hat{\boldsymbol{x}}_m(k|k-1)]^{\mathrm{T}}\} \cdot P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\} \quad (3.4)$$

Taken one step further, the hypothesis reduction algorithm will assign a probability weight, $P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\}$, to each newly created hypothesis during the measurement association and update cycle. Each measurement, $\boldsymbol{z}_j(k)$, associated with each pre-existing association history hypothesis, $\boldsymbol{\Psi}_u(k-1)$, creates a residual between the actual measurement and measurement predicted by the particular hypothesis' state, $\hat{\boldsymbol{x}}_u(k|k-1)$ defined as:

$$\boldsymbol{r}_{j,u}(k) = \boldsymbol{z}_j(k) - \mathbf{H}\hat{\boldsymbol{x}}_u(k|k-1) \quad (3.5)$$

This residual, as in the case for the standard Kalman filter, is Gaussian. In a non-MHT multiple-model estimator, each elemental Kalman filter creates only one measurement residual, and that residual is used in calculating that filter's modal probability. For the MHT tracker case, it will be necessary to create a pseudo-residual instead.

Since the residual produced by each pairing of a measurement and pre-existing component is Gaussian, the same assumptions made in Eq. (2.57) for the conditional pdf of the state within each elemental filter can also be applied to the conditional pdf of the pseudo-residual belonging to each elemental filter. The conditional pdf of the pseudo-residual produced by a Gaussian mixture of real residuals created within a hypothesis reduction filter based on a given dynamics model can be stated as:

$$f\{\boldsymbol{r}(k)|\boldsymbol{Z}^k\} = \sum_{u=1}^{N_h(k)} f\{\boldsymbol{r}(k)|\boldsymbol{Z}^k, \boldsymbol{\Psi}_u(k)\} P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\} \quad (3.6)$$

where $\boldsymbol{Z}^k$ is the measurement history through the current sample instant (k), $N_h(k)$ is the number of hypotheses created after associating all incoming and *gated* measurements with all $N_h(k-1)$ pre-existing hypotheses, $\boldsymbol{\Psi}_u(k)$ is one such association history hypothesis for $u = 1, \ldots, N_h(k)$, and $P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\}$ is the probability of the u-th hypothesis.

The pseudo-residual produced by each elemental filter m will have mean $\hat{\boldsymbol{r}}_m(k)$ and error covariance $\mathbf{A}_m(k)$:

$$\hat{\boldsymbol{r}}_m(k) = \sum_{u=1}^{N_h(k)} \hat{\boldsymbol{r}}_{m,u}(k) \cdot P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\} \quad (3.7)$$

$$\mathbf{A}_m(k) = \sum_{u=1}^{N_h(k)} \{\mathbf{A}_{m,u}(k)+ [\hat{\boldsymbol{r}}_{m,u}(k) - \hat{\boldsymbol{r}}_m(k)][\hat{\boldsymbol{r}}_{m,u}(k) - \hat{\boldsymbol{r}}_m(k)]^{\mathrm{T}}\} \cdot P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\} \tag{3.8}$$

and these two terms will produce the $[\boldsymbol{r}_m^{\mathrm{T}}(k)\ \mathbf{A}_m^{-1}(k)\ \boldsymbol{r}_m(k)]$ quadratic term necessary for calculation of the modal probabilities. Note that the summation occurs over all new hypotheses *prior* to hypothesis reduction, hence the $N_h(k)$ upper limit as opposed to the post-reduction $N_r(k)$ value.

Finally, the post-reduction analogue to $\hat{\boldsymbol{x}}(k|k)$ is calculated exactly as in Eq. (3.3) using the post-reduction hypotheses, $\boldsymbol{\Psi}_u(k)$ for $u = 1, \ldots, N_r(k)$, as components rather than the pre-reduction hypotheses. Note that this pdf involves the reduced number of components that result from the output of a hypothesis reduction filter.

One complication does arise in the pseudo-residual calculation of Eq. (3.6). Each filter in the MHT algorithm creates, at every time step, a *missed-detection hypothesis*. A missed-detection hypothesis will have $\hat{\boldsymbol{x}}_m(k|k) \equiv \hat{\boldsymbol{x}}_m(k|k-1)$ and $\mathbf{P}_m(k|k) \equiv \mathbf{P}_m(k|k-1)$, because this hypothesis assumes no measurement from the true target was actually recorded by the sensor, and so no measurement update was possible. The probability of a missed detection is derived in Appendix C and can be expressed as:

$$P\{\Psi_0(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} = P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}(1 - P_d P_g)\lambda \tag{3.9}$$

where $P\{\Psi_0(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\}$ is the joint probability of a missed-detection and $\boldsymbol{\Psi}_u(k-1)$ being the correct association hypothesis history at the time instant immediately prior to receiving the measurements at time instant (k) conditioned on knowledge of the measurement history through the (k)-th sample time, and $P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}$ is the probability assigned to the pre-existing hypothesis, $\boldsymbol{\Psi}_u(k-1)$, conditioned on the measurement history through $(k-1)$. The terms P_d, P_g, and λ, are parameters defined for the particular simulation; they represent the probability of target detection, the probability of the target-originated measurement falling within the measurement gate, and the clutter density (i.e., the expected number of clutter detections per unit hypervolume in the measurement space), respectively. It should be noted that, unlike Eq. (3.6), the conditional pdf of the target state described in Eq. (2.58) is indexed from 0 through $N_z(k)$, properly accounting for the missed detection hypothesis.

As shown Eq. (3.9), as P_d and P_g become smaller and as λ becomes larger, the probability assigned to the missed-detection hypothesis at each time step becomes increasingly large. Since the sum of all probabilities of all hypotheses in a mixture must always equal 1.0, a larger probability

on the missed-detection hypothesis implies that all newly created hypotheses, $\{\Psi_j(k), \boldsymbol{\Psi}_u(k-1)\}$ for $j = 1, \ldots, N_z(k)$ (where N_z is the number of measurements arriving at instant (k)) get *lower* probabilities. A missed-detection hypothesis, however, creates *no measurement residual* by definition. Consequently, the summations in Equations (3.7) and (3.8), which involve only real residuals between pairings of measurements and hypotheses, have improperly scaled probability weights.

One potential solution would be to rescale the probabilities of the positive-detection hypotheses so that they sum to one, thereby ignoring the weighting effect of the missed-detection hypothesis when computing the pseudo-residual. Another approach might involve the creation of a quantity that functions as the missed-detection term in the pseudo-residual computation. One such idea would be to introduce a single-sample memory in the $\boldsymbol{r}_m(k)$ and $\mathbf{A}_m(k)$ computation, such that the missed-detection term is the pseudo-residual and pseudo-residual covariance from time step $(k-1)$. In the pseudo-residual probability-weighted summations at sample time (k), these memory terms would be multiplied by the missed-detection hypothesis probabilities computed at (k).

The first approach seems slightly more consistent with the underlying concepts in a Kalman filter-based MMAE/IMM (i.e., the modal probabilities at sample (k) are always a function of an actual measurement residual at sample (k)), but the second approach does not disregard the impact a missed detection has on the total Gaussian mixture. Unfortunately, in that latter single-step memory technique, the $\mathbf{A}_m(k-1)$ term itself was calculated using a blending of component $\mathbf{A}_{m,u}(k-1)$ terms for each component $u = 1, \ldots, N_h(k-1)$, and each of those terms were calculated as $[\mathbf{H}\mathbf{P}_{m,u}(k-1|k-2)\mathbf{H}^{\mathrm{T}} + \mathbf{R}]$ using the u-th component's error covariance. While one could conceive of propagating these terms forward from $(k-1)$ to (k) in the single-step memory method, justifying some sort of propagation for the $\boldsymbol{r}(k-1)$ term is much more troublesome. Chapter 4 will present some brief performance comparisons between these two options.

3.4 Track Loss Checks

The stated goal of "tracking a target" can be interpreted in a variety of ways, and so the appropriate measure of tracking performance is similarly subject to interpretation. One may be interested strictly in whether the true target's position is within some hypervolume surrounding the tracker's blended state estimate position. Perhaps the boundaries of this hypervolume are determined by the blended error covariance, or perhaps it is determined by some predefined dimension. Maybe the application also requires that higher order states, such as velocity and/or acceleration, be within a certain range of the true target's velocity and/or acceleration. This would be the case if one were most interested in the ability to *predict* the target's position one sample period into the future, which is dependent on *current* estimates of position, velocity, acceleration, and possibly higher-

order derivative states. Maybe RMS miss distance, maximum/minimum miss distance, or mean miss distance in each axis are important criteria. For an MHT algorithm, the fact that at least one elemental filter within the MHT structure is maintaining lock on the target may be more important than whether the overall MHT blended state estimate is maintaining lock. That is to say, a single filter maintaining lock may allow for track recovery as more measurements are collected, and the deferred decision process of the MHT is allowed to operate, whereas the blended estimate may undergo large excursions from the true target location until that recovery occurs. Clearly, the metric used to declare whether a tracker is "tracking" the target may vary widely between applications.

3.4.1 Quadratic Quantity Metrics. One method of determining track loss is to evaluate a pure quadratic term $[\boldsymbol{x}^{\mathrm{T}}\mathbf{A}\boldsymbol{x}]$ for some vector $\boldsymbol{x}$ of dimension n and some n-by-n matrix $\mathbf{A}$. Such terms represent ellipsoids in n-dimensional space [29]. If the vector is formed as the difference between two renditions of the same vector, say truth values and filter-computed estimates from a simulation, and the matrix is the inverse of the error covariance of that difference, then this ellipsoid provides a multi-dimensional measure of the error committed by the filter's estimate compared to the filter-computed covariance of that error. This is precisely the type of operation a multiple-model tracker executes when calculating modal probabilities using the $[\boldsymbol{r}_m^{\mathrm{T}}(k)\ \mathbf{A}_m^{-1}(k)\ \boldsymbol{r}_m(k)]$ term.

Williams' research [30] used a quadratic metric for determining track loss on the single-dynamics-model hypothesis reduction filters he was testing. Specifically, it was formed for each association history hypothesis at every time step as

$$\boldsymbol{e}^{\mathrm{T}}\mathbf{P}^{-1}\boldsymbol{e} = [\boldsymbol{x}_{truth} - \hat{\boldsymbol{x}}_u]^{\mathrm{T}}\mathbf{P}_u^{-1}[\boldsymbol{x}_{truth} - \hat{\boldsymbol{x}}_u] \tag{3.10}$$

where $\boldsymbol{x}_{truth}$ is the full-dimensioned true target state, $\boldsymbol{x}_u$ is the full-dimensioned filter computed state estimate for the u-th association history hypothesis, and $\mathbf{P}_u$ is the associated error covariance for the u-th hypothesis. This allowed determining whether *at least one* filter (based on a specific association history hypothesis) was maintaining track. In addition to using this residual on *every component*, Williams also used it on the *overall Gaussian mixture* as developed in Eq. (2.57) [1]. This allowed the algorithm to judge whether any or all of the maintained hypotheses are maintaining track as well as whether the entire mixture is maintaining track.

The pure quadratic is an excellent means of measuring the overall tracking ability of the algorithms, and it will be retained in this research. However, as stated earlier, some applications may find this to be too general. One option is to compute the quadratic using fewer states, thereby

[1] Williams used the $\hat{\boldsymbol{x}}_u(k|k)$ state and $\mathbf{P}_u(k|k)$ terms rather than the $\hat{\boldsymbol{x}}_u(k|k-1)$ and $\mathbf{P}_u(k|k-1)$ terms described in Eq. (2.57) (see Section 3.3.2)

removing the effects caused by having a very good or very poor estimate in states of little interest. For some applications, perhaps position estimates are the only quantity of interest, and this modification would be justified.

3.4.2 Position Prediction Capability. A novel method of determining track loss might be to measure how well a tracker "predicts" the position of the target one time step into the future. Assuming the target's motion stays relatively constant over a short sampling period, the metric might be derived as a quadratic function of the prediction error

$$e_{predicted}(k) = \boldsymbol{x}_t^{(pos)}(k+1) - \left[\hat{\boldsymbol{x}}^{(pos)}(k) + \hat{\boldsymbol{x}}^{(vel)}(k) \cdot T_s + 0.5 \cdot \hat{\boldsymbol{x}}^{(acc)}(k) \cdot T_s^2\right] \tag{3.11}$$

where $\boldsymbol{x}_t^{(pos)}(k+1)$ is the true target position one sample period into the future, $\hat{\boldsymbol{x}}^{(\cdot)}(k)$ is the tracker computed position, velocity, and acceleration at the current sample period (k), and T_s is the tracker sample period.

If this metric were evaluated post-simulation over several sample periods, it would give an indication of how well the tracker predicted the future position of the target. This metric has a good physical interpretation that is, perhaps, more physically meaningful than the quadratic shown in Eq. (3.10). There is potential utility in constructing both $[\boldsymbol{e}_{predicted}^{\mathrm{T}}\ \boldsymbol{e}_{predicted}]$ as well as $[\boldsymbol{e}_{predicted}^{\mathrm{T}}\ \mathbf{P}_{predicted}^{-1}\ \boldsymbol{e}_{predicted}]$, where the former is essentially a mean squared prediction error and the latter is the mean squared error compared to the tracker's own estimate of the second moment of that error. The incorporation of the filter computed covariance may illuminate tuning problems in the filters, since the covariance is essentially pre-computable for all time based on the specified filter dynamics. However, as will be shown in Chapter 4, if $\mathbf{P}_{predicted}$ grows inordinately large, it may skew any conclusions drawn from the $[\boldsymbol{e}_{predicted}^{\mathrm{T}}\ \mathbf{P}_{predicted}^{-1}\ \boldsymbol{e}_{predicted}]$ calculation, and $[\boldsymbol{e}_{predicted}^{\mathrm{T}}\ \boldsymbol{e}_{predicted}]$ may be a better alternative.

One point to note is the multiplication of the velocity by the sample period and the acceleration by the square of the sample period will cause those states to be weighted differently depending on the simulation sample period. Note as the sample period grows beyond 1.0, the velocity and acceleration have more pronounced effects than the position itself. Conversely, short sample periods emphasize the tracker's current position estimate over the acceleration and velocity estimate. In effect, this metric shows that long sample periods make the position and velocity states critical in determining future target position. Although this metric is not commonly seen in other tracker research, it provides insight into tracker performance that is difficult to obtain from the other metrics mentioned here.

3.4.3 Scalar Statistical Measurements Computed From Separate x-y States. The output of Monte Carlo simulations will provide blended estimates as well as individual filter estimates for each run. Furthermore, these simulations could record running estimates for every hypothesis generated over the course of the simulation. The only limitation is data storage space and the ability for the engineer to digest the information in a meaningful way. Since truth data is always available, it is often useful to evaluate the actual error committed by each state in each axis separately ($[\hat{\boldsymbol{x}}^{(i)}(k) - \boldsymbol{x}^{(i)}_{truth}(k)]$ for all states $i = 1, \ldots, n$).

From the actual error committed, it is possible to draw a variety of statistical measures. Root-mean-square (RMS) error, maximum or minimum error, mean error, actual error covariance or standard deviation (as well as computed), etc. These metrics are physically meaningful to the engineer, and they do provide a concrete quantifier for evaluating algorithm performance. Unfortunately, they can be somewhat misleading, because a real system doesn't necessarily benefit from such a separation of states or a statistical measure of the error committed in those states. For example, an absolute error of 2 meters in the x-position and 2 meters in the y-position provides a true miss distance of $2\sqrt{2} = 2.8284$, whereas a 1 meter error in the x-position and 3 meter error in the y-position produces a miss distance of $\sqrt{10} = 3.1623$. One engineer might look at the separate axis measurements and draw one conclusion about performance, while another engineer might draw a different conclusion after looking at the hypotenuse miss distance.

Regarding the meaningfulness of these statistical measures, a designer working on a target tracker for a hit-to-kill missile might be most concerned with the maximum miss distance at some point in time. This would directly impact the missile's lethality. On the other hand, the designer of an air-traffic control radar may be more concerned with the RMS value of the miss distance. Simply knowing the target is relatively close to the "blip" on the radar screen for a certain length of time might be sufficient for aircraft deconfliction. In summary, scalar statistics computed in the x and y axes independently are highly application-specific and are open to a high degree of interpretation. They will not be highly emphasized in this research.

3.4.4 Conclusions on Track Loss Checks. Because the conventional quadratic error measure of track loss (i.e., one using a "truth estimate minus filter or component estimate" error term) has such widespread acceptance in the tracking and estimation community, its importance cannot be ignored and will be considered in performance evaluations for this research. However, early experimentation showed that these quadratics applied either within an individual elemental filter or applied to the overall MMAE/IMM blended estimate declared track loss rather prematurely using a loss-threshold value of 10. While the $[\boldsymbol{e}^{\mathrm{T}}\, \mathbf{P}_e^{-1}\, \boldsymbol{e}]$ term's magnitude might indeed be larger than the chosen threshold for declaring track loss, it was obvious that the estimates either from the filters or

the blended solution were still quite good. Furthermore, it quickly became obvious that the presence of a single tracking filter (regardless of its quadratic term value) greatly helped any non-tracking filters during track recovery later in the simulation. An intelligently chosen quadratic term threshold, larger than the one used here, would probably alleviate the inconsistencies.

The physical motivation of the track prediction technique, as well as its ease of post-simulation computation, make it of primary interest. Because this measure of track loss is not commonly used, it is uncertain how best to apply it and present the results in a meaningful way. Appendix C will include sections in which the two quadratic track loss checks (the "conventional" track error and the track prediction error) are applied to specific algorithm configurations. These configurations will be chosen so that the differences and relative merits of the track loss checks can be quickly analyzed.

Scalar statistical measures, in general, are most appropriate as comparisons between configurations using a sufficiently large number of Monte Carlo runs with identical random number generator seed values for each set of runs (each Monte Carlo run set corresponding to a particular configuration). Unfortunately, they can become highly skewed by any runs in which the tracker had a divergent estimate (i.e., an unrecoverable loss of track). For this reason, the design process may involve: a) analysis of individual Monte Carlo runs to verify algorithm functionality and robustness, followed by b) analysis of track loss statistics computed from many Monte Carlo runs to determine overall algorithm track performance. Furthermore, the statistical measures are most meaningful if specific design specifications are established (an absolute maximum miss distance, for example). As discussed above, these measures could be computed from the separate x-y states, but this makes interpretation slightly more ambiguous and will not be emphasized in this research.

3.5 Ad Hoc Design Choices

Some non-MHT and MHT multiple-model tracker parameters must be chosen in an *ad hoc* fashion. As described in Chapter 2, some of these parameters (probability lower bounds, divergence bounds, Markov probability transition matrices, etc.) are inherent to the MMAE and/or IMM algorithms themselves. Other parameters, such as measurement gating probabilities or the desired number of components after Gaussian mixture reduction, are associated with the overall tracker or with the individual filters within the multiple model algorithms that form the tracker. Finally, some parameters, such as clutter density, probability of detection, or measurement noise strength, define the overall simulation environment and affect the actual information passing into the tracker. This section will attempt to address any design choices in such parameters, some of which were established based on accepted practice within the tracker community. Parameters without commonly accepted

values were chosen using sound engineering judgment, and an attempt will be made to explain the thought processes that went into those decisions.

3.5.1 Model Configurations. Chapter 2 described the motivation behind First-Order Gauss-Markov Acceleration (FOGMA) models and Thrust-Perpendicular-to-Velocity (TPV) models. In addition to those models, the constant-velocity model has been used extensively in target tracking research. Williams [30] chose to use a constant-velocity model in his work, and that model will also be included in this research. A continuous-time constant-velocity (CV) model is described by the state equation

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ y(t) \\ v_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{3.12}$$

where $w(t)$ represents zero-mean, white Gaussian noise with statistics:

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\delta(t-t') \end{aligned} \tag{3.13}$$

Given the three chosen models for target dynamics, FOGMA, TPV, and CV, a variety of truth/filter combinations can be tested. This research relied on several underlying concepts for deciding which combinations to test. In short, the following scenarios were chosen to illustrate particular attributes of the multiple-model trackers. Note that "truth" and "filter" models represent either FOGMA or CV unless specifically stated to be TPV.

- A single elemental filter running against a matching truth model with no purposeful maneuvers. This provides a best-case scenario. In a real-world application, this setup would show the best possible tracker performance in both the non-MHT (Kalman filter-based) and MHT (hypothesis reduction filter-based) cases.
- A single elemental filter running against a matching truth model with TPV maneuvers implemented by the truth target. This case truly illustrates the need for a multiple-model design, because a FOGMA or CV filter will have some difficulty tracking the maneuver if extreme.
- A single elemental filter running against a mismatched truth model. This represents a slightly more benign case than the TPV maneuver, but it should highlight the disadvantage of having only one dynamics model in a real-world environment in which the true target dynamics are uncertain.

- A single elemental filter running against a mismatched truth model with a maneuver created by a changing truth model over time (for example, a FOGMA with $T = 4$ switches to a FOGMA with $T = 1$ and a higher $\mathbf{Q}$ value). Again, this is a benign case compared to a TPV maneuver, and it highlights the inadequacy of a single filter model.
- A multi-filter system with a combination of CV or FOGMA filters against a matching, single truth model. This should highlight any performance degradation created by multiple models in a benign truth environment (in which multiple models are not necessarily beneficial).
- A multi-filter system with a combination of CV or FOGMA filters against a matching, single truth with a changing truth maneuver. This should highlight some of the advantages of multiple-model estimators, but will also showcase some of the inherent tuning difficulties when multiple, stochastic filters are pitted against stochastic truth models that change over time. Of particular interest is the time and tuning required for "in-force" filter models to "hand-off" probability weight to a more correct filter as the truth model undergoes changes in its dynamics model.
- A multi-filter system with one or more FOGMA/CV filters and multiple, mirrored, TPV filters. This represents the most all-encompassing multiple-model design given the assumptions made in this research about which models are realistic in a target tracking problem. This filter configuration will provide the best adaptability to a host of truth trajectories, and it will be run against all previously mentioned truth scenarios. However, this filter configuration will be the most computationally intensive and will mandate a host of state and covariance re-sizings of the form described in Section 3.2 at each time step (since a variety of state dimensions will be represented in the whole system).

3.5.2 MMAE Ad Hoc Parameters. As stated in Chapter 2, the MMAE is often configured to implement probability lower bounds on each elemental filter modal probability (to enhance probability flow) as well as upper bounds on the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ quadratic term (to provide an indication of filter divergence and need for restarting). Experience in tuning filter/truth configurations similar to those mentioned in Section 3.5.1 has shown that a probability lower bound between 0.01 and 0.001 is sufficient to keep "out of favor" filters from developing such extremely low probabilities that they will not recover quickly enough if a target maneuver causes the filter-truth match to change.

The MMAE divergence bound is slightly more difficult to pinpoint, because even during complex maneuvers, there is no guarantee any of the filters are indeed divergent. Filter divergence should occur very infrequently in a well-tuned system unless one or more filters are purposely unstable (to

allow rapid probability flow during transitions between truth dynamics models). TPV filters, while not unstable *per se*, may tend to diverge after some length of time when the truth model is benign. Given the presence of TPV filters, a divergence bound of 20 to 30 provides a reasonable cut-off for deciding whether such a filter's estimate should be reset to the current blended estimate.

3.5.3 IMM Markov Probability Transition Matrix . While the IMM may lack the *ad hoc* probability lower bound and divergence bound of the MMAE, its Markov probability transition matrix must be chosen in a somewhat *ad hoc* manner. As discussed in Chapter 2, insight into the real world may give an indication of the overall characteristics of the matrix, but insight does not necessarily lead directly to realistic *values* within the matrix.

For this research, three Markov matrices were used. The first case, used strictly for algorithm verification, is an identity matrix. Chapter 2 explains the equality of MMAE and IMM under an identity Markov matrix assumption. The other two configurations will be referred to as *nontransition-favoring* and *specific-model-favoring*. The actual matrices used in this research vary with the particular models used in the IMM, and a comprehensive list will appear in Appendix 2. This section merely explains the general architecture.

The nontransition-favoring matrix leads to a system that discourages model transitions between measurement updates. It can be viewed as a minor perturbation from an identity matrix. Such a system would exhibit many of the performance characteristics of a non-switching model (like MMAE) while still allowing some cross-flow between model states and covariances during the mixing phase. A nontransition-favoring matrix for a three-filter system might look like

$$\mathbf{T}(k|k-1) = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

A nontransition-favoring matrix has appeared in published research on IMM trackers in the past [3], so it will also be used in this research.

The specific-model-favoring matrix, on the other hand, implies an actual knowledge of which model matches truth, and the mixing process will tend to weight most heavily the filter believed to be most representative of truth (in an *a priori* sense). In a real-world system, the tracker designer may have only a slight notion of what the true target dynamics are, but a well-evaluated hunch might improve performance significantly. Furthermore, in a maneuvering target scenario, truth dynamics indeed change with time. The specific-model-favoring matrix leads to a system that tends always to pull towards the self-assumed "truth" matched filter, so it assumes any changes to actual truth are only temporary deviations. It does not entirely discourage model switching away from truth, hence

the small but non-zero terms in non-truth rows. A three-filter specific-model-favoring matrix with filter two assumed truth might look like

$$\mathbf{T}(k|k-1) = \begin{bmatrix} 0.15 & 0.15 & 0.15 \\ 0.7 & 0.7 & 0.7 \\ 0.15 & 0.15 & 0.15 \end{bmatrix}$$

Notice that all columns sum to 1.0, which implies the total probability of transitioning from a given model in the system to any model in the system (including itself) is unity, which is fundamental to a Markov model assumption. Variations to the specific-model-favoring matrix could be made by adjusting the proportions given to each row. For instance, perhaps one filter is most often truth, another filter is occasionally truth, while a third is only rarely truth. The truth-favoring matrix probability weight would be assigned in descending order based on the assumed proportion of time the system remains in each of the three modes.

A fourth option that was not explored is an *equally likely* Markov matrix. Such a matrix would have all elements of the same value with every column (also row, in this case) summing to 1.0. This would lead to a system that switches models without any presumption of what transitions are most likely. Yet another unexplored option would be a sequenced matrix allowing models only to transition in a particular order (as, only to a nearest neighbor in model parameter space). While such a matrix might be useful in systems that do not involve physical vehicles (an estimator for a communications application, for example), there is no fundamental reason why an aircraft or missile must transition in this way.

Finally, it is conceivable that the Markov transition matrix is *time-varying* in nature, with elements as a function of time since the most recently declared maneuver change. This would be appropriate for a manned aircraft under attack, since pilots are trained to pull a high-g turn for about 3-4 seconds, and then roll in order to accomplish an out-of-plane maneuver to be held for another 3-4 seconds, etc. If it is a short time since the last declared maneuver change, transitions are *unlikely*, and a nontransition-favoring form is appropriate. Transitions become *highly* likely about 4 seconds after the last maneuver change declaration. The time-varying matrix option presents a wide degree of Markov model flexibility, but it also puts greater emphasis on characterizing any time-varying or adaptive behavior of targets.

Clearly, the Markov matrix allows many degrees of freedom in determining model transition behavior. If model switching were primarily a function of the Markov matrix, the choice of Markov matrix values would be of extreme importance to algorithm performance. However, as is the case for the non-switching MMAE, the most accurate indicator of model transition will occur once the measurement residual is formed and the modal probability is evaluated. With that in mind, the

effect of a *poorly* chosen Markov matrix may be negated by the fidelity and efficiency of the modal probability calculation. At the same time, a *well* chosen Markov matrix may be of little value given that the measurement update accomplishes the same task. Finally, while both the MMAE probability and divergence bounds and the IMM Markov matrix are chosen in an *ad hoc* fashion, the MMAE bounds are only executed with the algorithm as needed. In that respect, the MMAE algorithm is not forcing an *ad hoc* model switch at every sample period using potentially inaccurate or stale assumptions. This research is designed, in part, to explore the practical effects of these design choices and compare their perceived benefits.

3.5.4 Hypothesis Reduction and Measurement Environment Parameters. Given that this research is simulation-based, there is a necessary connection between the choice of measurement environment parameters and hypothesis reduction parameters. The heavier the clutter density, the greater should be the number of components maintained in the Gaussian mixture after reduction. In a real-world tracker, the measurement environment is a function of the "outside world" as well as the particular sensor being used, so these parameters are largely fixed, or at the very least, difficult to change (e.g., the noise reduction in a radar receiver). The hypothesis reduction parameters will always remain part of the tracker, and they would essentially be independent variables in the real-world. In this research, as in any simulation environment, there is a need to balance computational burden with a realistic representation of the tracker's performance in a real-world environment. Consequently, these parameters are closely linked and will not necessarily provide equal performance in all environments.

In Williams' research [30], the ISE algorithm's primary independent parameter was the number of components in the Gaussian mixture leaving the reduction cycle. The compromise between performance and computational complexity is obvious. Under any measurement clutter scenario, more post-reduction components (in theory) allow the MHT to discern a target track better as time progresses and improve the fidelity of the blended Gaussian mixture state estimate. Furthermore, a large number of components makes the job of hypothesis reduction somewhat easier because less effort is exerted in reducing the high-order mixture to some sub-optimal lower-order mixture (i.e., a reduced-order mixture with more components requires fewer "tough" choices during reduction). However, a larger number of components after reduction does increase both computer memory requirements and more importantly, increases the number of components that must be propagated during the filter propagation cycle and associated/updated during the filter measurement update cycle.

For this research, most scenarios carried 30 components out of the mixture reduction cycle. This number was based on experience during the research by Williams [30]. In his research, as clutter

density grew substantially, his component number increased to a maximum of 50 components leaving the reduction cycle. In the multiple model algorithms, however, 50 components in each filter would imply a total of $(50 \cdot N_f)$ components leaving all reduction algorithms in the entire system containing N_f elemental Williams filters. The increased demand of propagating and updating these components was considered more detrimental than the increased demand inside the hypothesis reduction.

The parameter which most affects simulation execution speed is, by far, the clutter density. Since each predicted target location in each elemental filter produces a measurement gate[2], the total gating area defined for a particular filter is the union of individual gates [30, pg. 2-37]. Thus, the total number of measurements processed by the system is both a function of clutter density and the number of filters. A three-filter system must process three times as many measurements as a single-filter system under the same clutter density parameters (assuming no overlap between gates belonging to different filters and equally sized gate areas within each filter). Furthermore, as will be discussed in more detail in Chapter 4, the unioning of measurement gates between filters further magnifies the effect. The expected number of measurements per cycle is given in [23] as:

$$E\{N_z\} = \lambda \left(2 \cdot \mathrm{CF} \cdot \sqrt{R}\right)^s \tag{3.14}$$

where λ is the clutter density (expected number of measurements per unit hypervolume of the measurement space), R is the value of the s-th diagonal term in the measurement noise covariance, and s represents the dimension of each measurement (always 2 for this research). The "Clutter Factor" term CF can be controlled as an independent variable and allows the number of clutter measurements per cycle to be set at run time. Williams ran various clutter factors, the highest of which yielded about 480 measurements per cycle and allowed him to compare the ISE-optimization-based reduction algorithm performance to work conducted by Salmond. That degree of clutter caused substantial, but acceptable, computational burden for Williams' single-dynamics-model simulation. However, when accounting for the overhead of the multiple model algorithms themselves, plus the fact that each elemental filter is running its own instance of a Williams ISE reduction, that level of clutter was quickly dismissed as too high to be computationally feasible. Instead, two baseline clutter cases were considered. A "low-clutter" case yielding about 5 measurements per cycle, and a "medium" clutter case yielding about 10 measurements per cycle were considered. From an engineering judgment standpoint, these numbers seemed like a realistic representation of the number of measurements a real sensor (such as a modern radar) would provide a tracking algorithm during normal operation. For diagnostic purposes, an "ultra-low" clutter density case, yielding an expected

[2]Each component in a Gaussian mixture represents a predicted target location assuming a state propagation using that filter's dynamics. This motivates a separate gate for each component rather than one gate for the entire filter.

2 measurements per cycle, was considered for purposes of verifying the MHT algorithm functionality in an environment very close to that seen by the Kalman filter-based algorithms. A special discussion of this ultra-low clutter density case will appear in Chapter 4, Section 4.3. Chapter 4 will also explain in further detail how measurement gate unioning magnifies the effect of clutter density.

Finally, the effect state dimensionality has on clutter parameters should be briefly noted. As shown throughout the preceding developments, as clutter density or mixture-component carry-over number increases, the number of states and covariances to be propagated and updated clearly increase in proportion across cycles. A constant velocity dynamics model involves a four-state estimate and a sixteen-element error covariance computation for every component in the Gaussian mixture (10 elements of which must be computed if symmetry is exploited). A FOGMA model, on the other hand, requires a six-state estimate and a *thirty-six*-element error covariance (with 21 non-redundant elements). So, increasing the dynamics model dimension by a single state in both axes (x and y) represents a 50% increase in state estimate dimension over a constant-velocity filter and a 125% increase in the total number of error covariance elements. The effect this dimension increase has on an MMAE/IMM MHT is further multiplied by the number of filters of that dimension in the overall system. Compared to Williams' research, which concentrated on a single-dynamics-model system using a four-state constant-velocity model, the burden of a even a two-filter system using two six-state FOGMA's represents a considerable increase in computational effort. That effect weighed heavily on the choices of models and clutter parameters used in this research.

3.6 Summary

This chapter began by showing how partition-sensitive modal and mixing probabilities allow filter and truth models of different state dimensionality to exist within the same tracking algorithm. The technique maximizes the use of information available within the system during filter resets (MMAE) and initial condition mixing for each propagation/update processing cycle (IMM) by passing higher-order derivative states when they are available and truncating these states when they are not needed (as would be the case for lower-order filters in the system). It also permits the tracker blended estimate to have a state-dimensionality equal to the highest-order filter in the system rather than the lowest-order filter.

A detailed discussion of the steps necessary to migrate the Kalman filter-based multiple-model algorithms to Williams filter-based MHT multiple-model algorithms followed. The concept of pseudo-states, pseudo-state error covariances, pseudo-residuals, and pseudo-residual error covariances was introduced. It was shown that MMAE or IMM elemental filters based on Gaussian mixtures can produce these quantities as probability-weighted sums of the individual mixture component state es-

timates and error covariances. It was also shown that pseudo-residuals can be formed as probability-weighted sums of actual residuals once gated measurements have been associated with a filter's pre-existing set of components.

Various track loss checks were evaluated with the intent of using some of them in the analysis of simulation results. The conventional definition of track loss, computed as either $[\boldsymbol{e}^{\mathrm{T}}\ \boldsymbol{e}]$ or $[\boldsymbol{e}^{\mathrm{T}}\ \mathbf{P}^{-1}\ \boldsymbol{e}]$, is a measure of the absolute state estimate error committed by the tracker at every sample period. A similar measure is the error committed in the *prediction of the target position one sample period into the future*, computed as either $[\boldsymbol{e}^{\mathrm{T}}_{predicted}\ \boldsymbol{e}_{predicted}]$ or $[\boldsymbol{e}^{\mathrm{T}}_{predicted}\ \mathbf{P}^{-1}_{predicted}\ \boldsymbol{e}_{predicted}]$. This was considered somewhat more physically meaningful than the conventional quadratic measures because it relied upon modeling assumptions (namely propagation period) in determining the weight applied to each state within a state estimate. Finally, track loss statistics computed from separate x-y states were considered, but the difficulty of their interpretation along with the lack of any concrete design criteria will make them of less interest in this research.

Lastly, the numerous *ad hoc* assumptions necessary to implement these trackers were addressed. A brief discussion of the various filter model and truth model combinations to be used in simulation trials was undertaken. The combinations are designed to highlight particular performance attributes of the MMAE and IMM algorithms, as well as provide realistic representations of real-world targets. The choices necessary in building a Markov probability transtion matrix for an IMM were analyzed. It was concluded that an identity matrix, a so-called nontransition-favoring matrix, and a specific-model-favoring matrix were of particular interest to this research. Finally, the factors involved in choosing measurement clutter density and the specification of the number of Gaussian components remaining in a mixture after reduction were addressed. In general, as clutter density increases, it makes sense to increase the number of components remaining after each Williams filter mixture reduction operation.

Applied together, the concepts in this chapter help to build multiple-model tracking algorithms using either Kalman or Williams filters within each elemental filter. Chapter 4 will show the results of simulation trials using these implementations.

IV. Simulation Results

Chapter 3 discussed the assortment of engineering decisions made during the development of the four tracking algorithms: 1) a Kalman filter-based MMAE, 2) a Kalman filter-based IMM, 3) a Williams-filter-based MMAE, and 4) a Williams-filter-based IMM. Chapter 4 will discuss the performance of these four algorithms during Monte Carlo trials, and it will analyze their relative performance from both a qualitative and a quantitative standpoint. Section 4.1 will show the performance of the Kalman-filter-based algorithms in a clutter-free environment, with the implication that performance of these algorithms will broadly represent the best performance one could hope to achieve using the Williams-filter-based MHT algorithms against targets in clutter. Particular attention will be paid to performance trade-offs between IMM and MMAE for the given scenarios.

After simulation trials began, several unanticipated shortcomings appeared in the MHT algorithms developed in Chapter 3. Section 4.2 will analyze these shortcomings and discuss modifications designed to alleviate the problems. Some of these modifications greatly affect computational speed and the ability of the MHT to maintain track, so their importance cannot be overlooked.

Sections 4.3, 4.4 and 4.5 will analyze the performance of the Williams-filter-based MHT algorithms as implemented using the modifications discussed in Section 4.2. Aside from the incorporation of measurement clutter, the simulation scenarios used during MHT testing were the same as those used during the Kalman-filter-based algorithm testing, and performance comparisons will be made.

Throughout Chapter 4, a standard set of plots will be used to illustrate performance attributes. Appendix A explains the content of the various plot styles and how to interpret the information for a given scenario. Appendix B lists the different filter model combinations (to be referred to as suites), the various truth trajectories (to be referred to as scenarios), as well as the various IMM Markov probability transition matrices used in this research. A short qualitative description accompanies each suite and scenario listing to give the reader insight into that suite or scenario's physical motivation. When specific data is presented for analysis, a note will be made referring the reader to Appendix B for more detailed information about the suite and scenario of interest.

Finally, all simulation sets discussed in this chapter are comprised of 10 Monte Carlo trials, each of which was run to 300-sample-times (the declared point of trial completion). In the section covering the medium-clutter trials, a few of those simulation sets comprise only 7 Monte Carlo runs, and this is noted in the analysis pertaining to those trials. In all other cases, the data represents the full set of 10 trials. All trials assume a 1.0 second (in simulation time) sample period and thus a 1 Hz (in simulation time) measurement update rate. Simulation units are assumed to be in meters (position), meters/sec (velocity), and seconds (simulation time).

4.1 Kalman-Filter-Based Results

This section will cover the results of simulations using the Kalman-filter-based algorithms for the case in which there is no measurement clutter. First, Section 4.1.1 will discuss how different choices of measurement noise covariance affect the performance of the multiple-model algorithms. The topics discussed in that section are relevant to all of the simulations conducted for this research. Sections 4.1.2 through 4.1.7 will analyze the Kalman-filter-based algorithm simulation performance for various filter suite/truth scenario configurations. Section 4.1.8 will discuss the performance of an IMM with a specific-model-favoring Markov transition matrix and will explain why this Markov matrix was dismissed from future study in this research. Finally, Section 4.1.9 will summarize the results of the Kalman-filter-based algorithms and draw some conclusions about relative performance of the various configurations.

The reader should note that the track loss checks described in Chapter 3, Section 3.4 will be applied where useful, but those results will appear in Appendix C. In general, the analysis presented in this chapter will give an excellent *qualitative* idea of algorithm track loss performance, and the track loss checks in the appendix simply provide more *quantitative* analysis.

4.1.1 Relation Between Algorithm Performance and Measurement Noise Covariance. It is important that the reader understand how the choice of measurement noise covariance affects the conclusions that one can draw from the performance data. In past tracker research, including that by Williams [30] and Salmond [23], a common choice for the measurement noise covariance was an identity matrix (i.e., the noise had a standard deviation of 1 "unit"). The choice of this value, however, cannot be separated from the assumed units used throughout the system. This research assumed all units within the system were based on 1.0 meter rather than a unitless dimension of 1.0. A measurement noise covariance of an identity matrix with units of meters implies the sensor has a measurement error standard deviation of 1.0 meter – a rather accurate sensor, indeed! The effect this has on multiple-model algorithm performance is significant.

As long as a filter's state estimate is reasonably close to the measurement at the end of a 1.0 second propagation cycle, the filter will not diverge. Because the measurement noise covariance is so low, the Kalman filter gain is rather high, the system greatly "trusts" the measurements, and the filter estimate will "snap" back towards the measurement. The filter will always catch up to the measurement after an update, and the filter dynamics become less of a factor. In the Kalman-filter-based algorithms, this leads to a great many combinations that perform reasonably well, and precise assessments about which suites work best for which scenarios becomes more difficult. In the MHT algorithms, this behavior will be present as well, but the presence of measurement-association uncertainty will dominate the effects of a low measurement noise covariance. In both Kalman-based

and MHT algorithms, a substantial increase in measurement noise would lower the Kalman gains, so proper choice of filter dynamics and *ad hoc* algorithm parameters would take on greater importance.

4.1.2 Single Filter Against Benign Truth. Under the assumptions of zero-mean, white Gaussian noise inputs and linear system models, a single Kalman filter will be the *optimal* estimator in any scenario in which a single dynamics-model filter matches the simulation truth dynamics-model for all time. Chapter 2, Section 2.2.1, discusses Kalman filters and optimality criteria in more detail. A single-filter model that matches the simulation truth model for all time is, therefore, the ideal target tracking configuration in the sense that no other tracker configuration (under the assumptions in this research) will provide a higher fidelity state estimate and error covariance. To evaluate the performance of a single Kalman filter, the Kalman-filter-based IMM was configured to use a single elemental filter with an identity Markov probability transition matrix. Although a dedicated, single filter algorithm would exhibit faster run times, the state estimate and error covariance from this special case IMM will be identical to that of a dedicated, single-dynamics-model Kalman filter. The use of the special case IMM was motivated only by the desire to limit the number of software configurations needed to implement all aspects of this research.

Figure 4.1 shows the output for the constant-velocity dynamics and Fig. 4.2 shows the output for the benign FOGMA case. The leftmost column of the elemental filter plots shows the mean $\pm 1\sigma$ x (azimuth) and y (elevation) position and velocity errors committed by the filter, taken as the mean and standard deviation over all Monte Carlo runs at the given sample time. This information would not be available in real-time usage. Superimposed on these graphs is the *filter-computed* zero $\pm 1\sigma$ values, represented as wide gray lines.

The first and second plots in the rightmost column show measurement residual data mean $\pm 1\sigma$. Again, this is taken across all Monte Carlo runs at the given time step. Unlike the state estimate error plots, the residual information is available on-line within the filter itself. Note that zero $\pm 1\sigma$ (filter-computed standard deviation) is superimposed on these plots as well, and it is represented as a wide gray line.

The third plot from the top in the rightmost column pertains to the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ value mean $\pm 1\sigma$ taken across all runs for a single filter (or elemental filter in an MMAE or IMM). Finally, the fourth plot in the rightmost column represents the filter's modal probability, which in the single-filter tracker cases, is always 1.0.

Notice that these configurations exhibit $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ quantities with magnitudes relatively close to the dimension of the measurement (always 2 in this research). Values greater than one

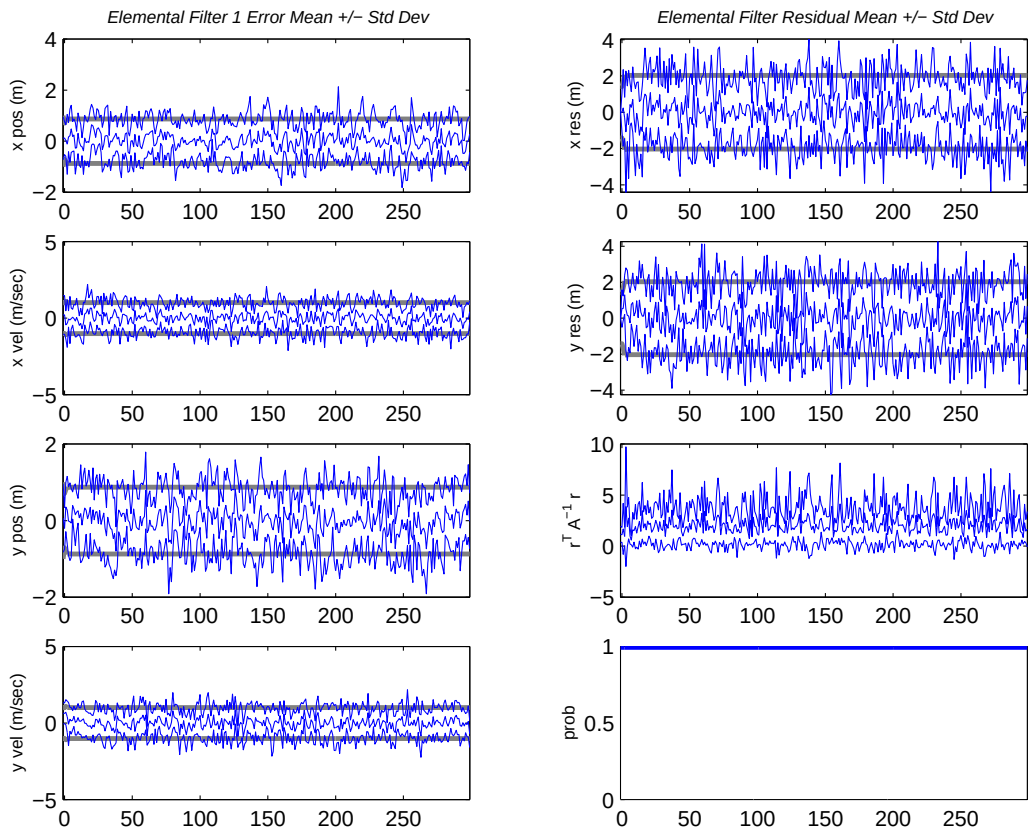

Figure 4.1: The single constant-velocity filter dynamics match the truth model dynamics for all time. (Suite 1 vs. Scenario 1: IMM (Identity Markov))

order-of-magnitude higher than the measurement dimension generally indicate that filter is a poor match for the truth trajectory. This criterium will be applied throughout the forthcoming analysis.

4.1.3 Single Filter Against Time-Varying Truth Model. Varying parameters of the truth dynamics-model presents a slightly more challenging scenario for the single-dynamics-model Kalman filter. Assuming the filter dynamics-model is not time-varying and is tuned to match a benign truth model, a single filter versus a time-varying truth model configuration will exhibit worse tracking performance than the case in which the filter dynamics match truth dynamics for all time. The degree of performance degradation is determined by the degree of dissimilarity between the benign truth model (i.e., the truth dynamics-model that matches the filter dynamics-model) and the "maneuvering" truth model (i.e., the truth dynamics-model that does not match the filter's dynamics).

In Fig. 4.3, the target is undergoing benign flight from ($k = 0, \ldots, 99$), executes a more aggressive FOGMA maneuver for ($k = 100, \ldots, 199$), and returns to the benign case for ($k = 200, \ldots, 300$).

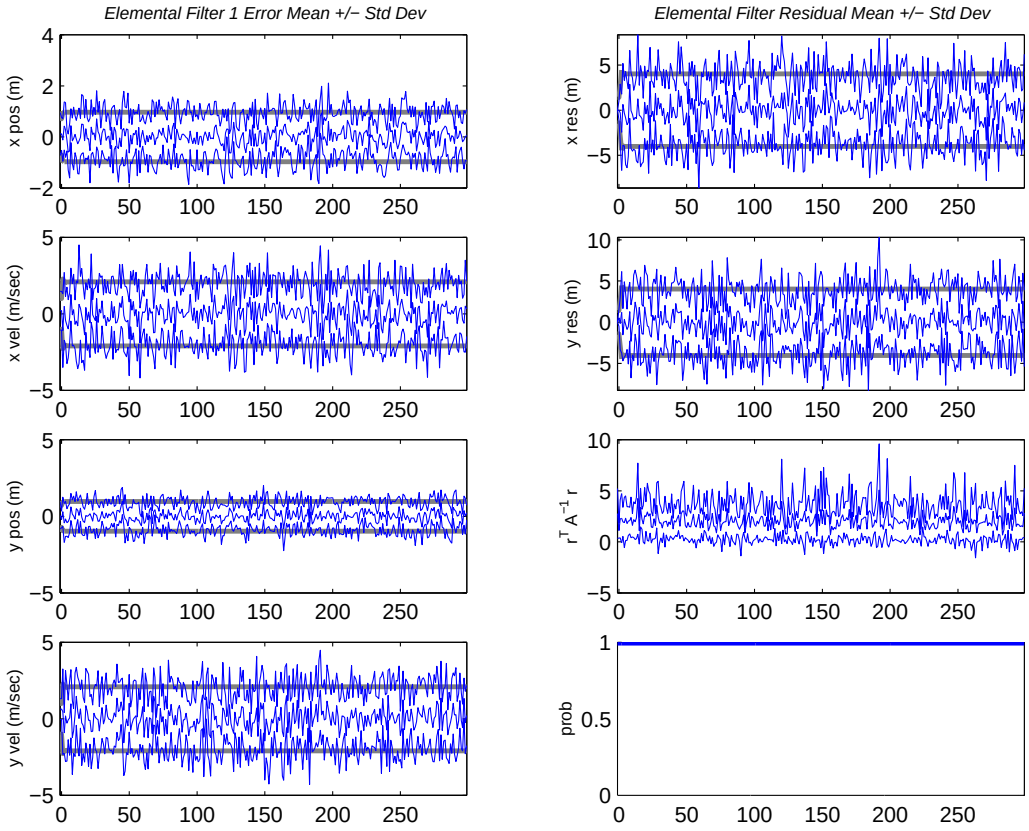

Figure 4.2: The single FOGMA filter dynamics match the truth model dynamics for all time. (Suite 2 vs. Scenario 2: IMM (Identity Markov))

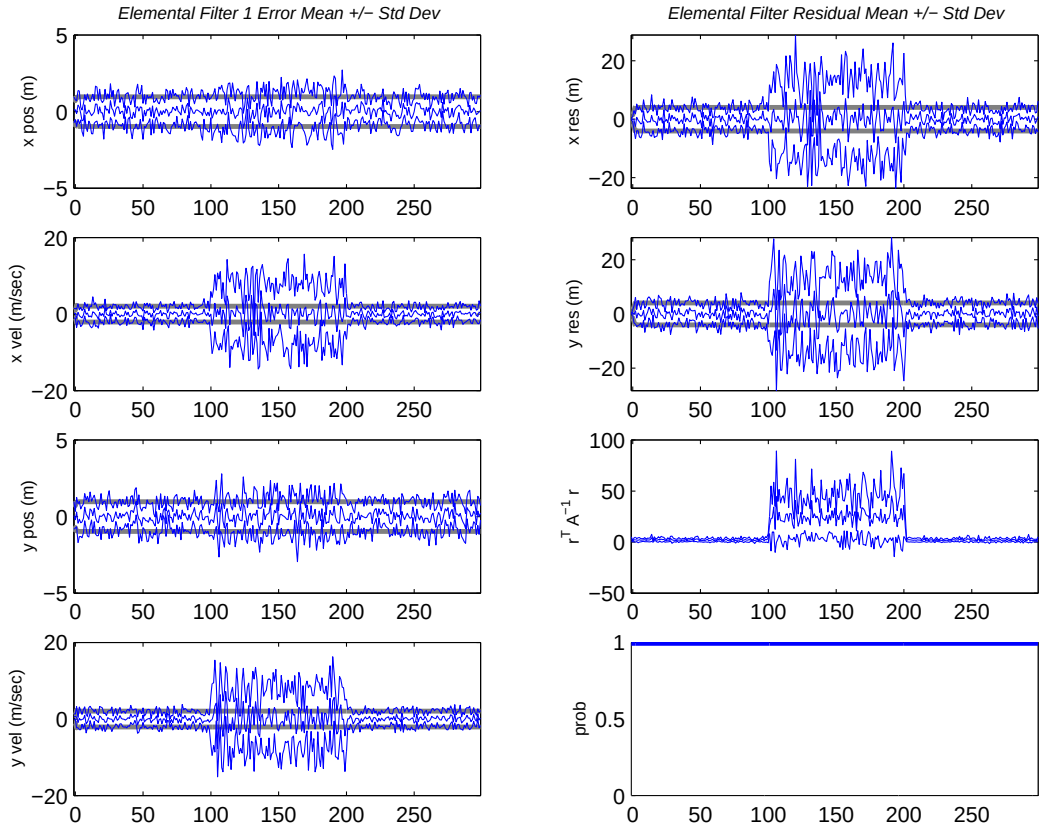

Figure 4.3: The single FOGMA filter provides a reasonably good estimate against a time-varying truth model composed entirely of FOGMA dynamics. (Suite 2 against Scenario 4: IMM (Identity Markov))

While the single-dynamics-model FOGMA filter tracked the benign portions adequately, it tracked the more aggressive truth dynamics less well. The $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ magnitudes become significantly larger than 20 during the maneuver, indicating the filter is a poor match to truth despite the fact that filter divergence never occurred (a result of the low measurement noise).

If the single filter were tuned for the aggressive truth dynamics, the results are somewhat better. Figure 4.6 is the output of filter 2 (matches the aggressive maneuver) from a two-model MMAE running against the same time-varying truth model of interest here. If this filter were the *only* filter in the system, the tracker output error would be represented by the position and velocity error plots in this figure's left column. Notice the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ mean plus 1σ never goes higher than 10 during the aggressive maneuver and is quite small during the benign truth model phases of simulation. In almost every respect, during the benign phases of simulation, the state estimate error committed by this filter is not noticeably worse than the single benign filter which matches

the benign truth. During the aggressive maneuver, this filter yields an estimate with *moderately less* error than the benign filter. The primary performance difference, though, is seen in the filter-computed covariances. The aggressive filter will overestimate the actual error committed during the benign phase, whereas the benign filter will underestimate the error committed during the aggressive maneuver phase. Again, if the measurement noise covariance were higher (in the real world as well as in the filter design model), both of these filters would have more difficulty maintaining lock during non-matching phases of flight.

A worse matching occurs when a benign constant-velocity filter is attempting to estimate a target modelled by the same aggressive FOGMA dynamics from above. In Fig. 4.4, the benign portion of the target's flight adhered to a more benign FOGMA model designed to approximate a constant-velocity model. Software limitations did not allow a time-varying truth model to have underlying dynamics models with differing numbers of states (e.g., a 4-state constant-velocity model could not change to a 6-state FOGMA model), hence a CV truth had to be approximated with a very benign FOGMA in this scenario.

Notice the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ magnitude jumps considerably during the maneuver – significantly more so than in Fig. 4.3. The plot in Figure 4.4 extends only to $(k = 190)$, because the filter diverged from truth on several of the Monte Carlo runs. Nonetheless, it is clear this single-filter suite is not versatile enough to handle the aggressive maneuver.

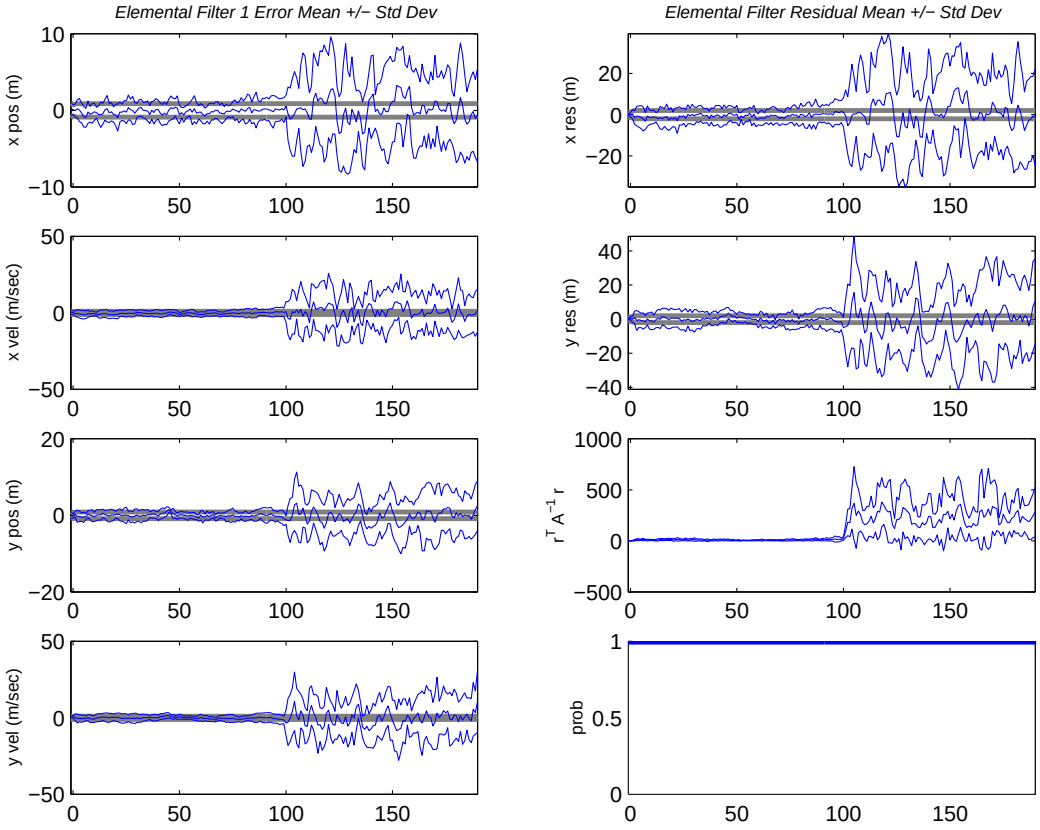

Figure 4.4: A CV filter against an aggressive FOGMA maneuver. (Suite 1 vs. Scenario 3: IMM (Identity Markov))

4.1.4 Configurations with Two-FOGMA Filters. The truth model scenarios considered in the previous sections are time-varying and purely stochastic (i.e., they have no deterministic or consistently nonzero-mean inputs, only zero-mean random process inputs). In addition, they are intentionally constructed such that two, piecewise time-invariant dynamics sets (each set consisting of a unique state transition matrix $\mathbf{\Phi}$ and dynamics driving noise matrix $\mathbf{Q}_d$) can represent all phases of the trajectory. A filter bank consisting of two filters, each with a dynamics model that match a particular truth model set, is the most logical tracker configuration, given what is known about the simulated truth. For the sake of discussion, this special two-model configuration will be called *piecewise matched* since, for all time, one of the filters matches the truth model in force at any given time.

Clearly, having complete knowledge of the truth model scenario (as is the case in the simulation software developed for this research) *does* artificially influence the choice of filter bank configurations. Nonetheless, past experience in configuring Kalman-filter-based MMAE trackers showed that multiple-filter banks that include more than one filter of a given type (e.g., three FOGMA filters all with slightly different q or T parameters) perform well but often result in diluted probability weights among all of the filters of the type. In other words, even if the truth model were continuously varying in time, as long as the underlying dynamics continued to obey some general form (i.e., a FOGMA with continuously varying q and T), a large bank of filters (each of the same type but with different parameter values) is not *necessarily* better than a small bank of filters (each of the same type but with different parameter values than those in the large bank).

Insightful tuning of the filter dynamics parameters, careful attention to the discretization of model parameters within the parameter space, and (in the MMAE case) careful use of *ad hoc* restart techniques upon declaration of a divergent elemental filter, will aid probability flow among filters in configurations involving multiple filters of the same general form. With that in mind, a thoughtful reduction in the number of such filters can alleviate many of the tuning and discretization difficulties, yet preserve much of the state estimation fidelity associated with systems containing a larger number of filters (of the same dynamics). Although the computational burden associated with additional filters in the Kalman-filter-based algorithms is mostly insignificant, the effect in the MHT algorithms is quite dramatic. Consequently, an attempt was made in this research to keep filter count as low as possible without unacceptably sacrificing tracker state estimation fidelity.

Suite 4, a tracker with two FOGMA filters, is piecewise matched to the truth trajectory in Scenario 4. Results for the MMAE are shown in Figs. 4.5, 4.6, and 4.7. Because the suite includes a filter matched to each segment of the time-varying truth model and no additional filters, this arrangement should exhibit very low state estimate error compared to most other filter bank

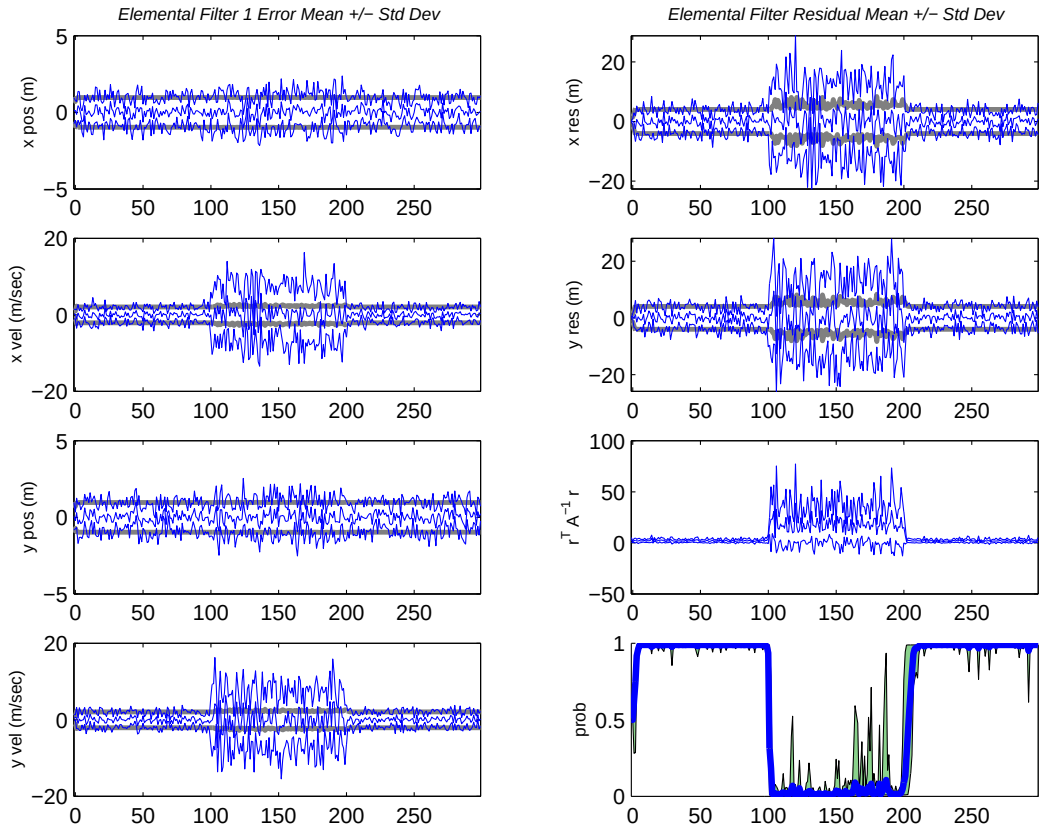

Figure 4.5: Filter 1, a benign FOGMA, from a two-filter system. This filter matches truth during the first and last third of the simulation. (Suite 4 vs. Scenario 4: MMAE)

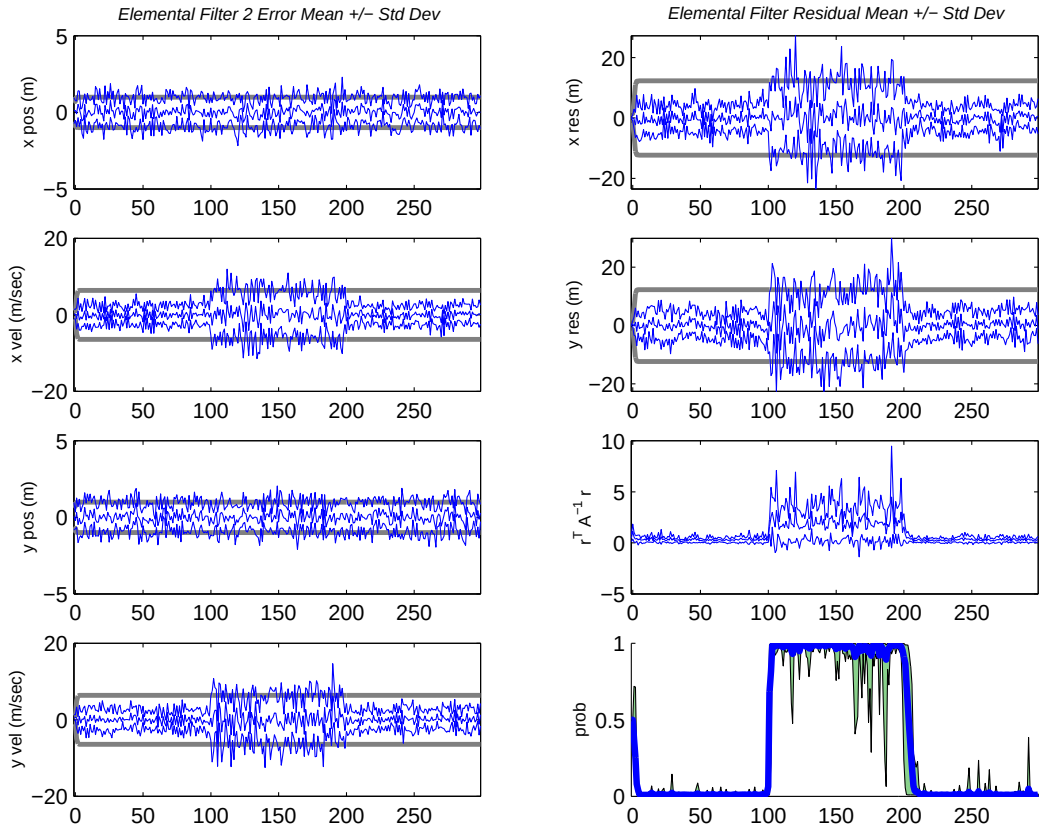

Figure 4.6: Filter 2, an aggressive FOGMA, from a two-filter system. This filter matches truth during the middle third of the simulation. (Suite 4 vs. Scenario 4: MMAE)

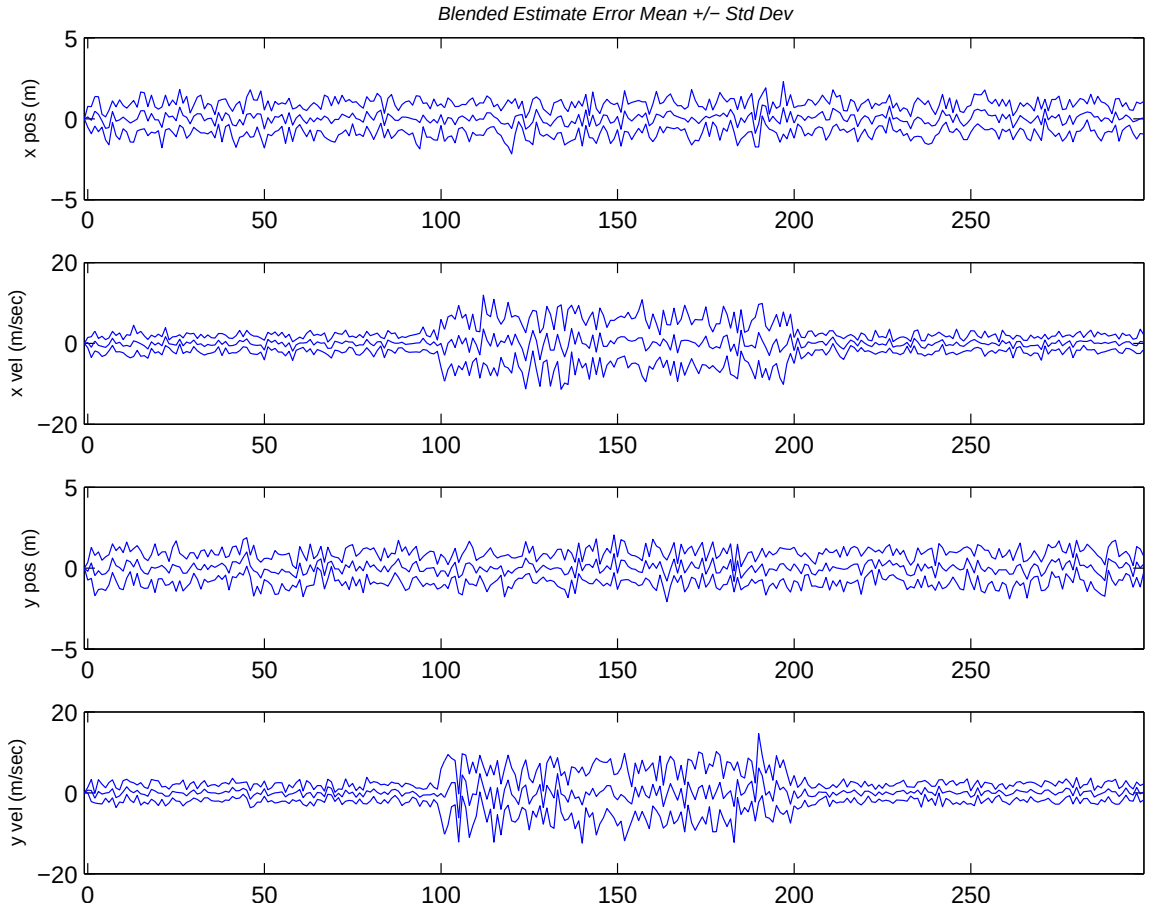

Figure 4.7: Blended estimate for the two-filter system. (Suite 4 vs. Scenario 4: MMAE)

configurations running against this truth scenario. Filter 1, the benign FOGMA, has output (shown in Fig. 4.5) that is essentially identical to that of the filter in Suite 2 (shown in Fig. 4.3), which has identical dynamics. The probability flow clearly shows elemental filter 2 taking the majority of the probability weight during the aggressive portion of the truth trajectory, and this gives an indication that the tracker does readily recognize the change in truth model dynamics.

Notice that the aggressive FOGMA Filter 2 in Fig. 4.6 has relatively small residuals during the benign phases (first 100 and last 100 seconds of the simulation). On one hand, this might lead to the conclusion that the filter dynamics parameter space is discretized too finely. On the other hand, overly fine discretization will normally cause ambiguous probability flow, which is not the case here. To analyze the ability of this suite's aggressive FOGMA to track a maneuver with a higher QF value[1], Scenario 7 was created. The FOGMA maneuver in this scenario has QF=4.0 rather than QF=2.0 as in Scenario 4, and the results for the MMAE are shown in Figs. 4.8, 4.9, and 4.10.

[1]The QF scaling factor appears in the driving strength covariance term and is discussed in detail in Appendix B.

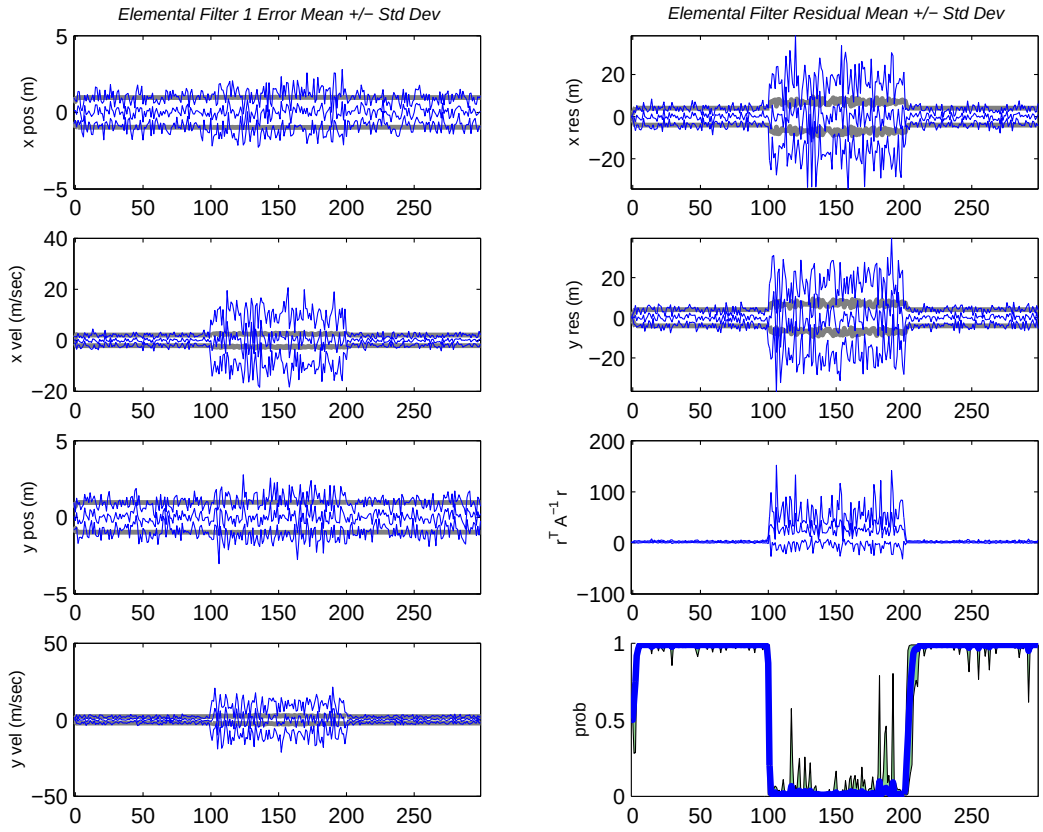

Figure 4.8: Filter 1, a benign FOGMA, from a two-filter system. This filter matches truth during the first and last third of the simulation. (Suite 4 vs. Scenario 7: MMAE)

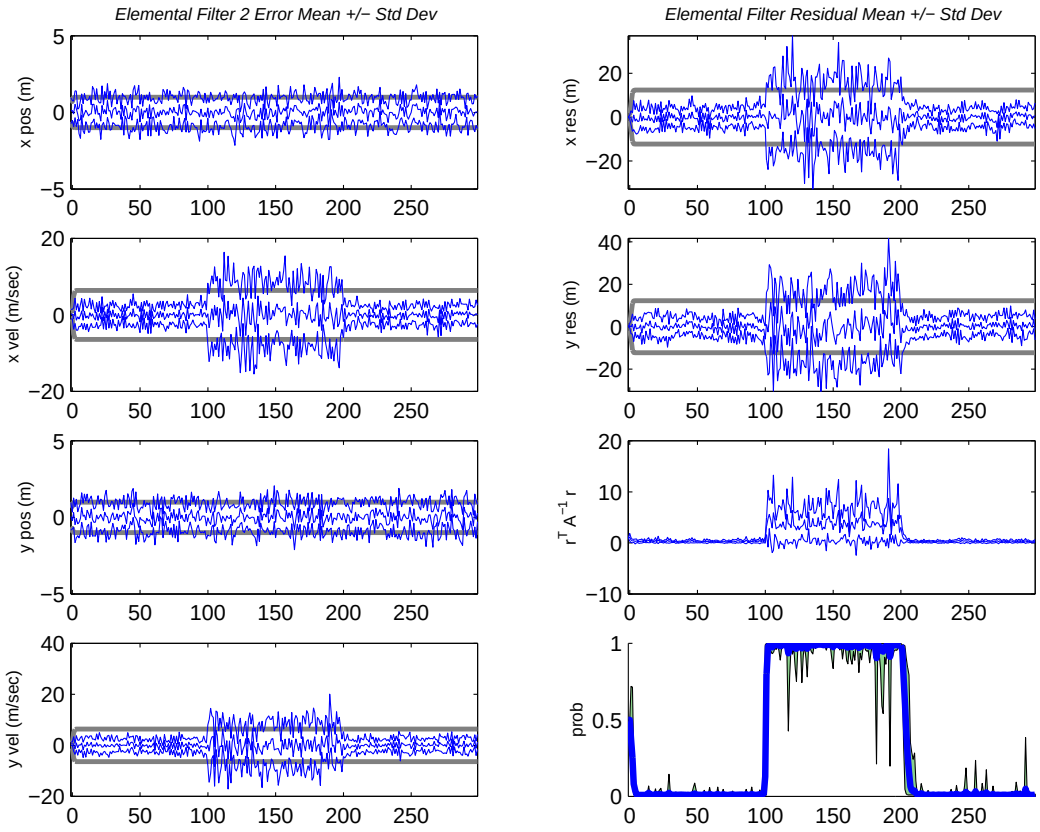

Figure 4.9: Filter 2, an aggressive FOGMA, from a two-filter system. During the middle third of the simulation, this filter has a q value only half as large as the truth q. (Suite 4 vs. Scenario 7: MMAE)

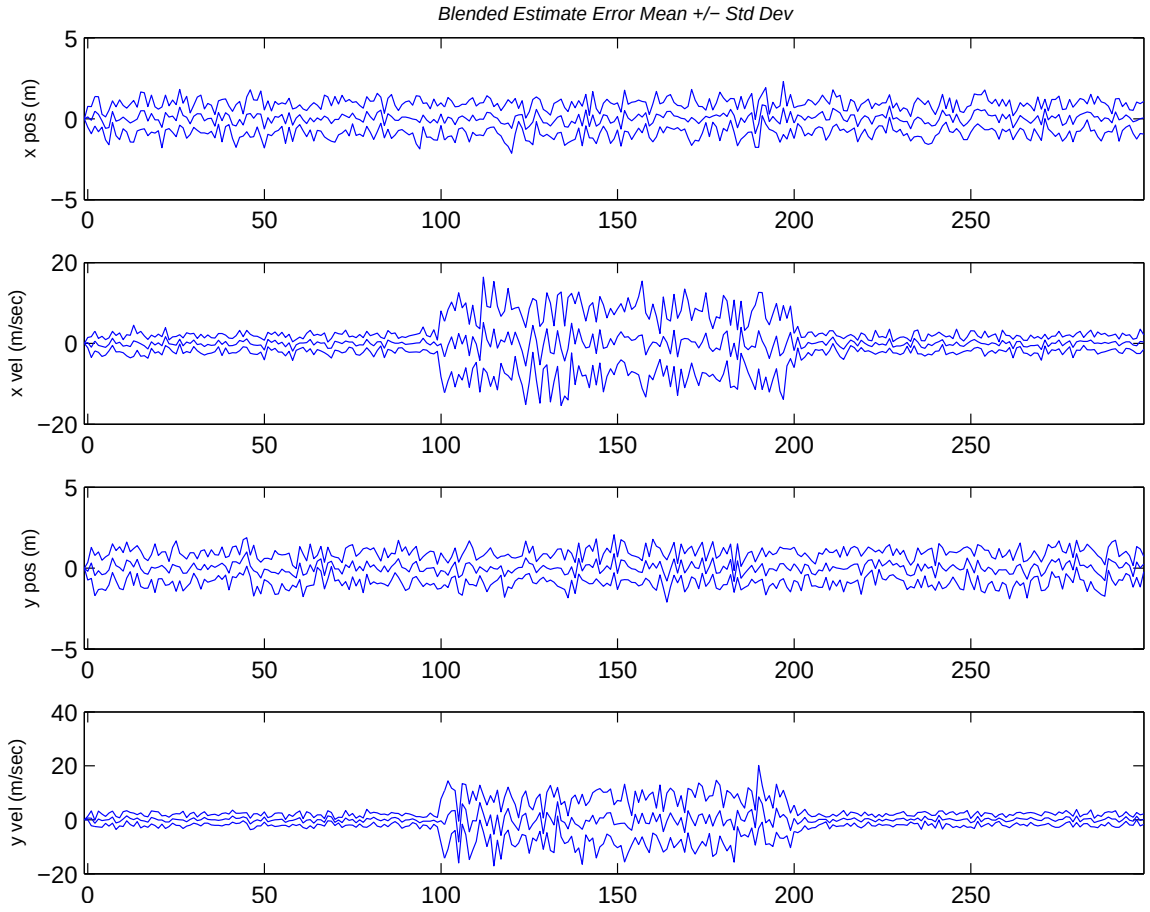

Figure 4.10: Blended estimate for the two-filter system. (Suite 4 vs. Scenario 7: MMAE)

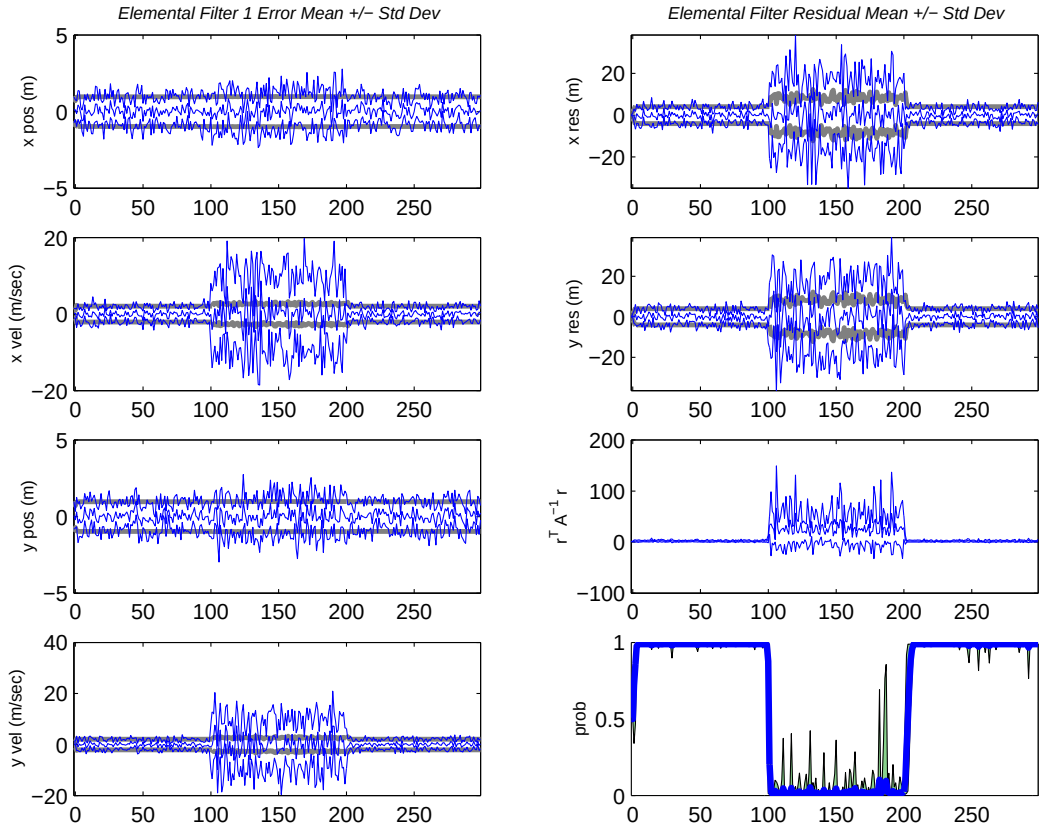

Figure 4.11: Filter 1, a benign FOGMA, from a two-filter system. This filter matches truth during the first and last third of the simulation. (Suite 7 vs. Scenario 7: MMAE)

Finally, Suite 7 was developed so that the second elemental filter precisely matched the truth model during the very aggressive FOGMA maneuver from ($k = 100, \ldots, 199$). This suite's filter 2 has a q twice as large as Suite 4's filter 2, but all other filter parameters are identical. In theory, the state estimate performance should be better for this suite than it was for Suite 4 when running against Scenario 7, which Suite 7 matches precisely. The elemental filter plot for filters 1 and 2 appear in Figs. 4.11 and 4.12. Notice that the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ magnitudes on filter 2 are better than they are when that filter has the smaller q (Suite 4). The blended estimate results are shown in Fig. 4.13, but the error committed by this suite's blended estimate is not noticeably different from that shown in Fig. 4.10 for Suite 4. This could be anticipated by the probability plots, elemental filter state estimation performance plots, and residual plots being essentially the same for the two cases. Chapter 5 will explain the motivation behind increasing the aggressive FOGMA driving noise strength by about one to two orders of magnitude in any future research.

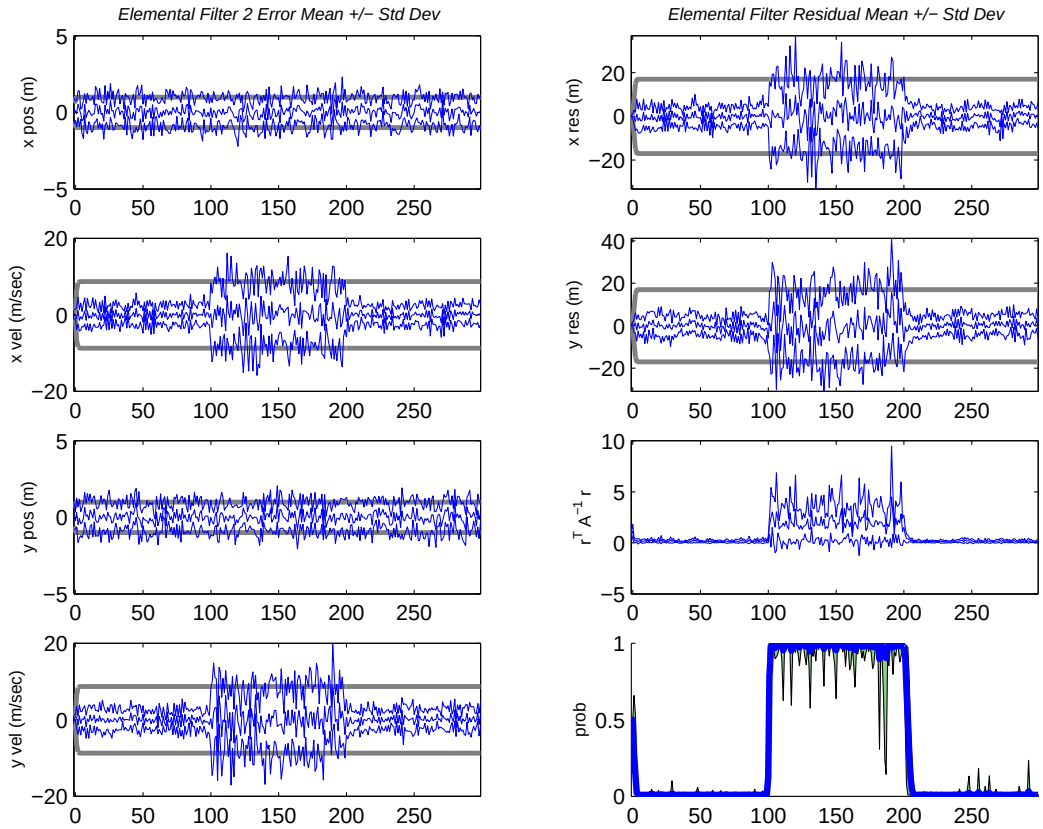

Figure 4.12: Filter 2, an aggressive FOGMA, from a two-filter system. During the middle third of the simulation, this filter has a q value equal to the truth q. (Suite 7 vs. Scenario 7: MMAE)

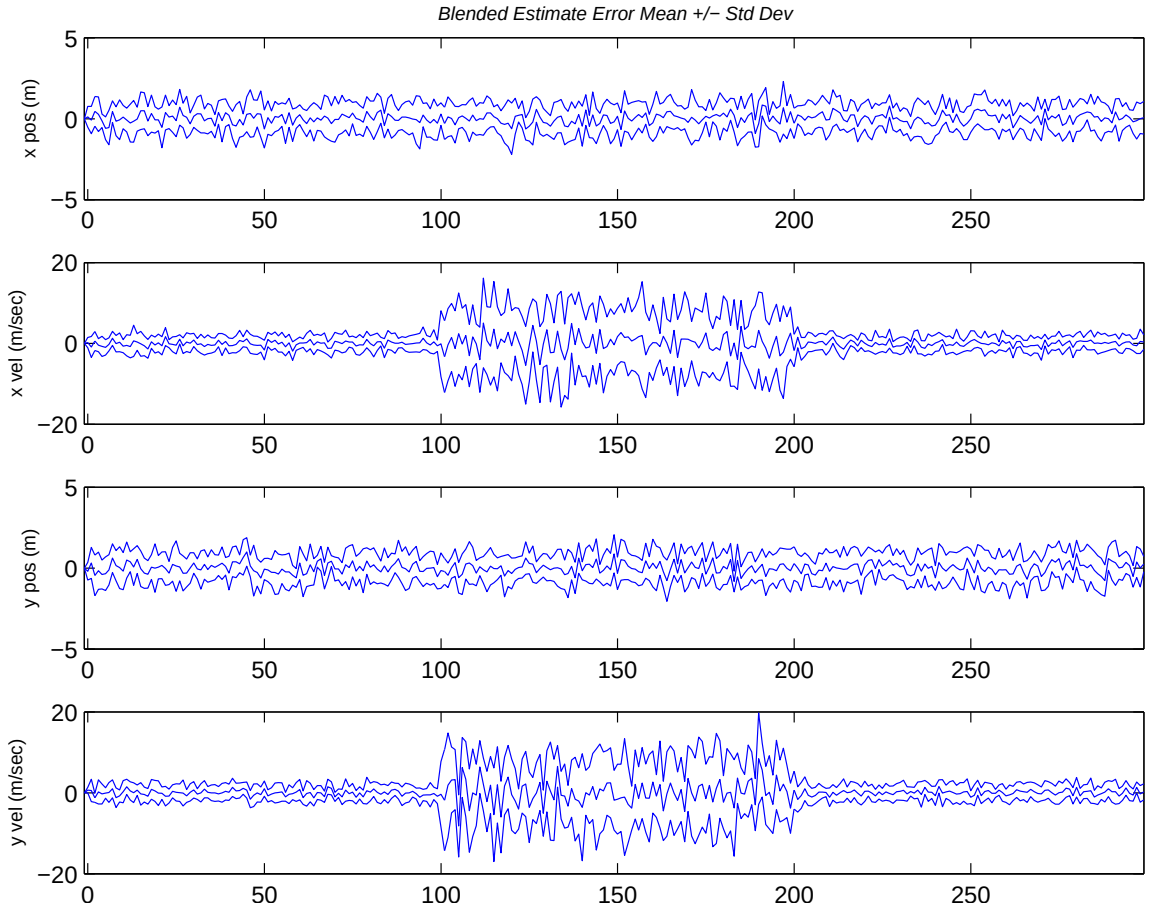

Figure 4.13: Blended estimate for the two-filter system. (Suite 7 vs. Scenario 7: MMAE)

The two-filter tracker simplifies the tradeoffs made in the discretization of the filter dynamics parameter space yet provides a relatively high-fidelity (small error) state estimate. The FOGMA models are relatively flexible in their ability to handle truth dynamics that are moderately more aggressive than the dynamics of the filter itself. Furthermore, the foreseen computational load in an MHT tracker resulting from two, 6-state filters in the filter bank should not tax the algorithm unnecessarily.

4.1.5 Single Filter Against Truth Model with TPV Maneuvers. Since most airborne targets spend a major portion of their flight time in a "benign" (or non-maneuvering) state, it is sensible always to include a benign filter within any multiple-model configuration. Constant-Velocity (CV) and long-period, low-noise FOGMA models can both function as benign filters. Chapter 2 justified the use of Thrust-Perpendicular-to-Velocity (TPV) filters to represent aggressive target maneuvers, because most airborne targets maneuver in directions nearly orthogonal to their current velocity.

Single elemental filters with constant-velocity or FOGMA dynamics perform decidedly worse against Thrust-Perpendicular-to-Velocity (TPV) maneuvers than they do against truth models with time-varying but entirely zero-mean stochastic models (i.e., no deterministic or persistently-constant-but-nonzero-mean input), as discussed in the previous sections. A convincing example of this poor performance is exhibited by a single CV filter matched against a CV truth model with three, +3-g TPV maneuvers, occuring at various times throughout the simulation (truth model Scenario 5). Performance under this scenario is shown in Fig. 4.14.

The FOGMA model handles the maneuvers a little better. In Scenario 8, the underlying truth dynamics are represented by a very-long-period, very-low-noise FOGMA, with the +3-g TPV maneuvers executed on top of this model at the times listed in Appendix B. Elsewhere, the truth model obeys a benign FOGMA matching the single elemental filter. From Figure 4.15, it is obvious the FOGMA handles the TPV maneuvers better than a constant-velocity model. However, the velocity estimate error plots show obvious correlation with the maneuvers, and the filter residuals spike at the beginning and end of the maneuvers.

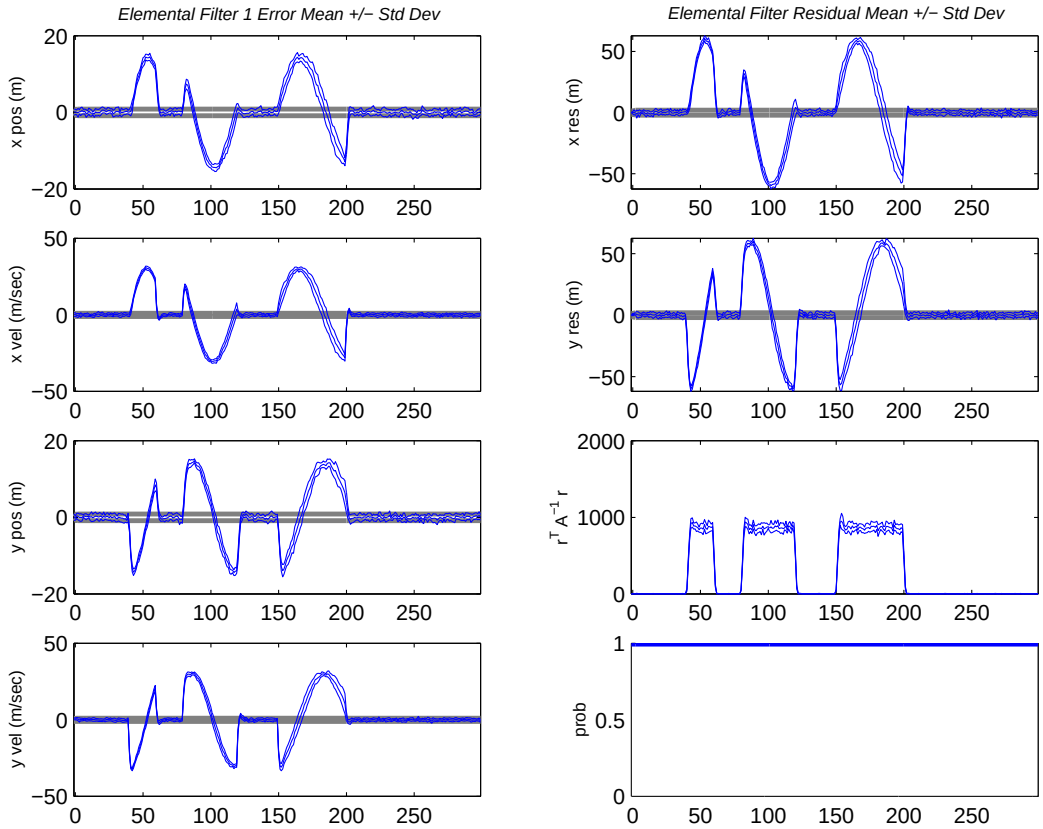

Figure 4.14: A CV filter against a truth model with underlying CV dynamics but executing three, +3-g TPV maneuvers. (Suite 1 vs. Scenario 5: IMM (Identity Markov))

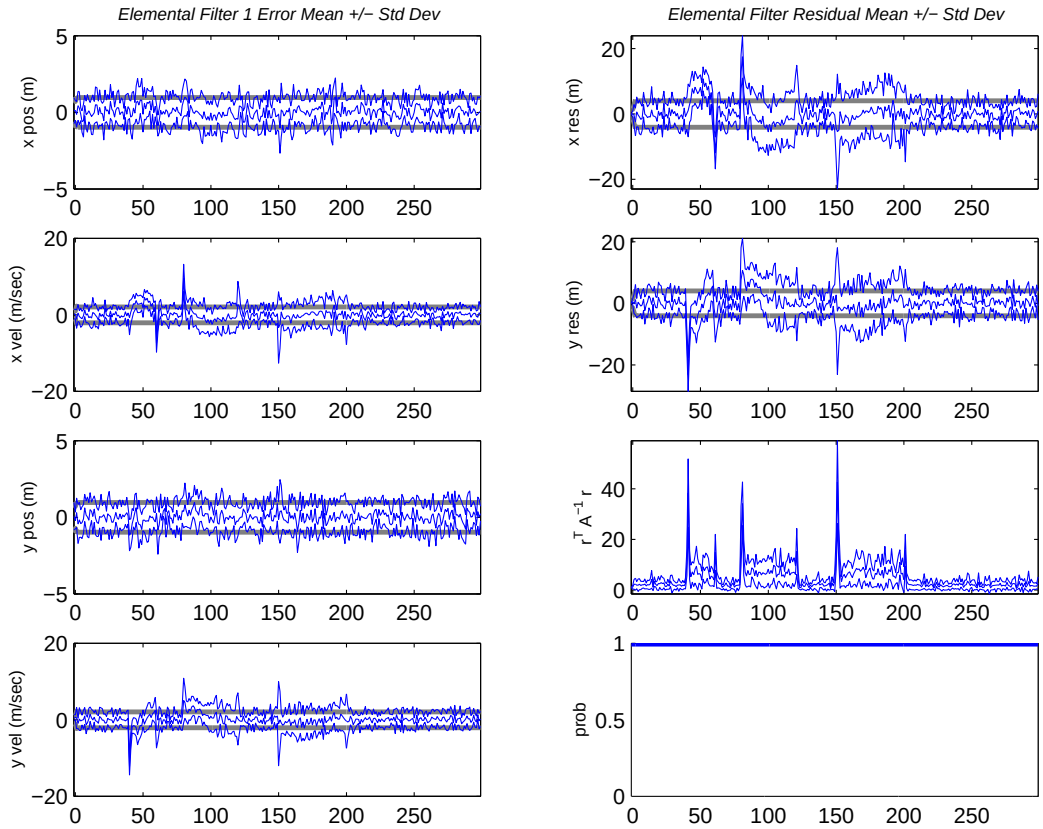

Figure 4.15: Single FOGMA elemental filter against a series of TPV maneuvers. (Suite 2 vs. Scenario 8: IMM (Identity Markov))

4.1.6 Suites with Two TPV Filters and Either One Constant-Velocity or One FOGMA Filter. Having shown the physical motivation for targets executing TPV maneuvers, and having seen the poor-to-mediocre performance of the single CV or FOGMA filter models against such maneuvers, inclusion of filters exhibiting TPV dynamics is very justifiable. TPV filters in this research were constructed with an underlying CV model with additional deterministic $\boldsymbol{B}_d(k)$ and $\boldsymbol{u}(k)$ terms. The CV dynamics are justified since a TPV maneuver, by definition, will be flown with an essentially constant-magnitude velocity even though its direction is changing over time. The $\boldsymbol{B}_d(k)$ term is calculated at each time step as discussed in Chapter 2. Although not shown explicitly in Eqs. (2.16) and (2.17), in this research, those terms are multiplied by the acceleration of gravity (i.e., yielding a 1-g turn) and a scaling factor so that TPV maneuvers can be implemented as multiples of 1-g maneuvers[2]. Finally, $\boldsymbol{u}(k)$ will be 2×1 matrix of all ones in elemental filters that assume a TPV maneuver, since the TPV filters assume the TPV maneuver (defined by the $\boldsymbol{B}_d(k)$ term) is in effect for all time.

TPV filters are most effective if the scaling factor on Eqs. (2.16) and (2.17) are "mirrored". That is, a TPV elemental filter designed for a positive 3-g maneuver should be mirrored by another TPV elemental filter designed for a negative 3-g maneuver. This is not to say a target is performing a negative 3-g maneuver in the traditional sense. Rather, the coordinate frame defining our sensor measurement space has a fixed Cartesian reference frame, and the positive or negative TPV maneuver merely defines which direction within the measurement space a target is moving. For example, a manned-aircraft moving in the positive x-direction in the measurement space but flying "inverted" (topside of aircraft facing the ground) can pull a +3-g maneuver (in the pilot's reference frame) that appears as a -3-g maneuver in the sensor reference frame (which has no indication of target roll-angle).

As discussed above, model parameter-space discretization is a consideration in choosing TPV filter parameters. Again, the discretization choices made in this research were driven by a need to keep the number of filters to a computationally manageable level yet maintain an acceptable state estimate error. TPV filters in a Kalman-filter-based multiple-model architecture incur relatively insignificant computational penalties, but within an MHT, computational effects are magnified. Each Gaussian component (each of which corresponds to a particular assumed measurement association history) in a TPV Williams filter within an MHT must undergo its own online $\boldsymbol{B}_d(k)$ computation – a substantial task when one considers each MHT filter may contain upwards of twenty components. With this in mind, the Kalman-filter-based implementations were restricted to one pair of mirrored

[2]Appendix B shows continuous-time dynamics representations for *all* models used in this research as they were adapted for use in the simulations. The adapted representations differ very little from those discussed in previous chapters.

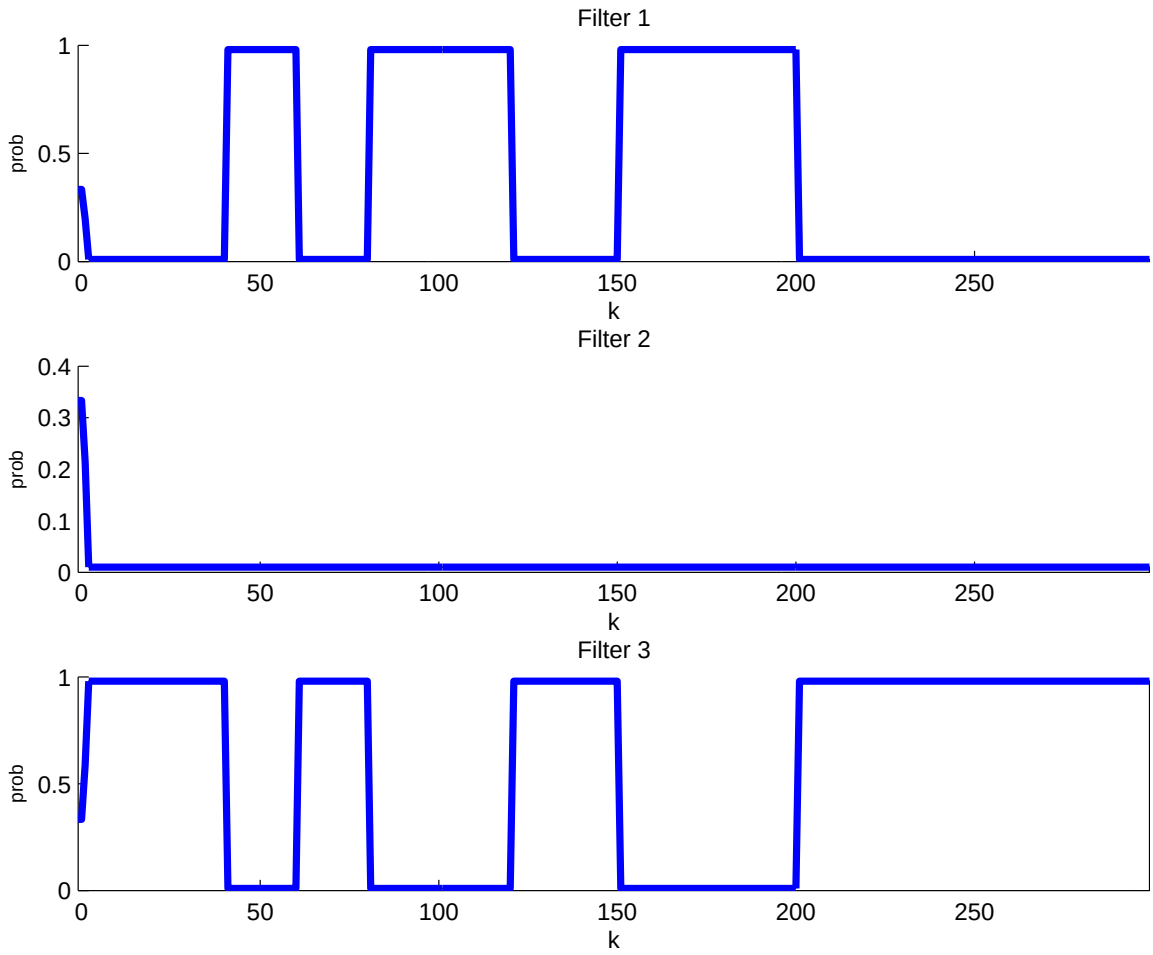

Figure 4.16: Probability flow for a filter bank in which filter 1 is a +3-g TPV, filter 2 is a -3-g TPV, and filter 3 is a CV. The truth model executes three, +3-g TPV maneuvers over the course of the simulation. (Suite 5 vs. Scenario 5: MMAE)

TPV filters, even though two or three pairs with appropriately spread g-loadings would be more representative of airborne vehicle maneuver capability.

Figures 4.16 through 4.20 illustrate how the addition of two, mirrored TPV filters improved the state estimate considerably compared to the CV filter alone (as depicted in Fig. 4.14). Notice that all blended position and velocity errors in Fig. 4.20 are much closer to zero-mean and much less time-correlated than they are in Fig. 4.14. The probability flow shows that the +3-g TPV filter is taking the probability weight extremely well during the maneuvers, and given the fact that an exact filter model match exists for every phase of the truth trajectory, this is to be expected. The spikes in the blended estimate represent an instantaneous (i.e., a single sample period) increase in state estimate error on the transitions between TPV maneuvers and the benign CV phases of truth flight

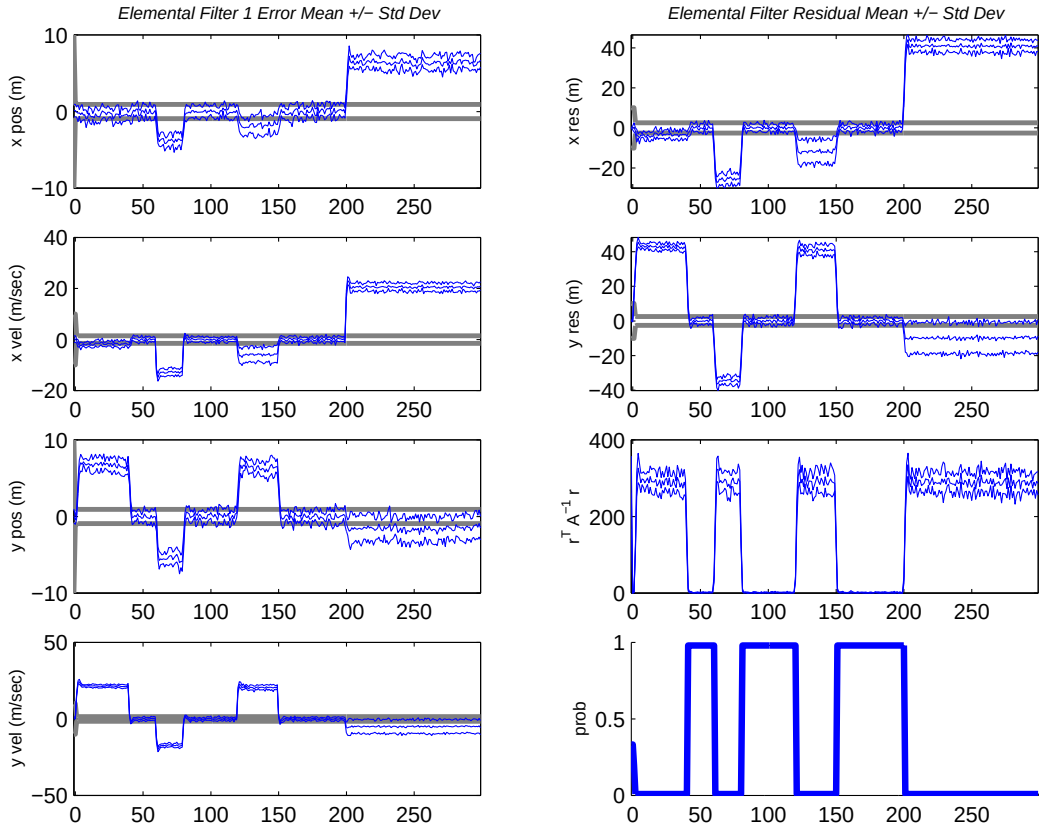

Figure 4.17: Filter 1 output. (Suite 5 vs. Scenario 5: MMAE)

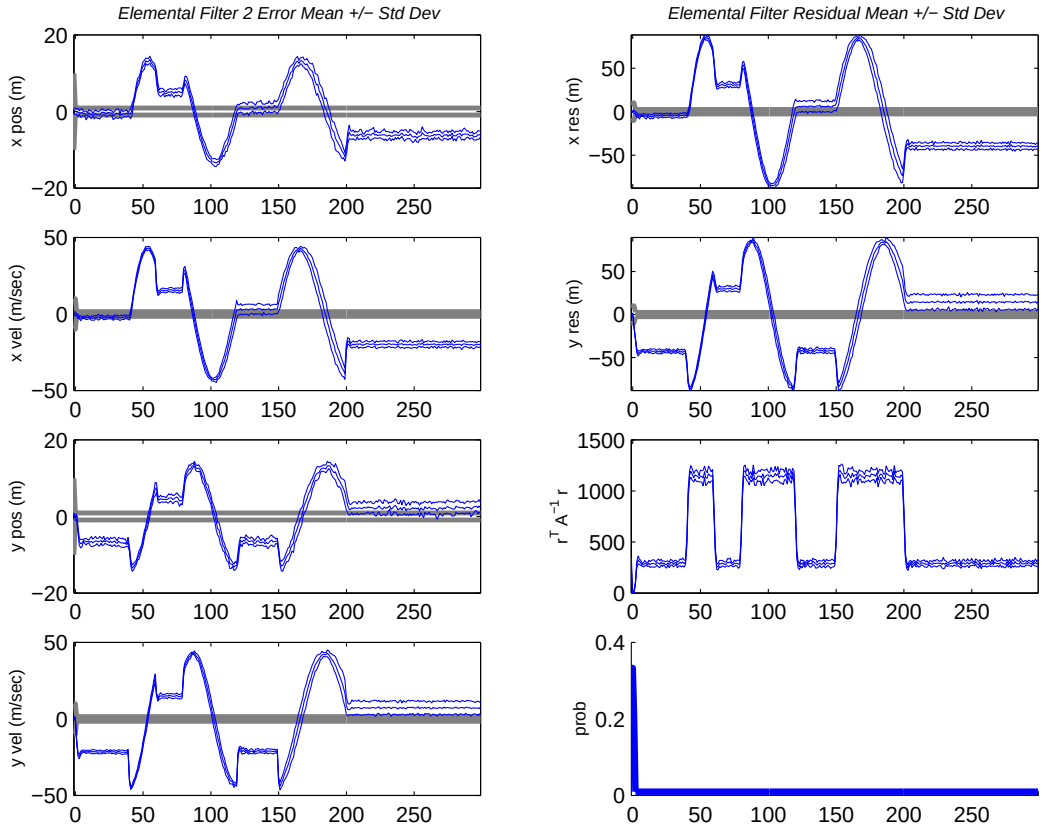

Figure 4.18: Filter 2 output. (Suite 5 vs. Scenario 5: MMAE)

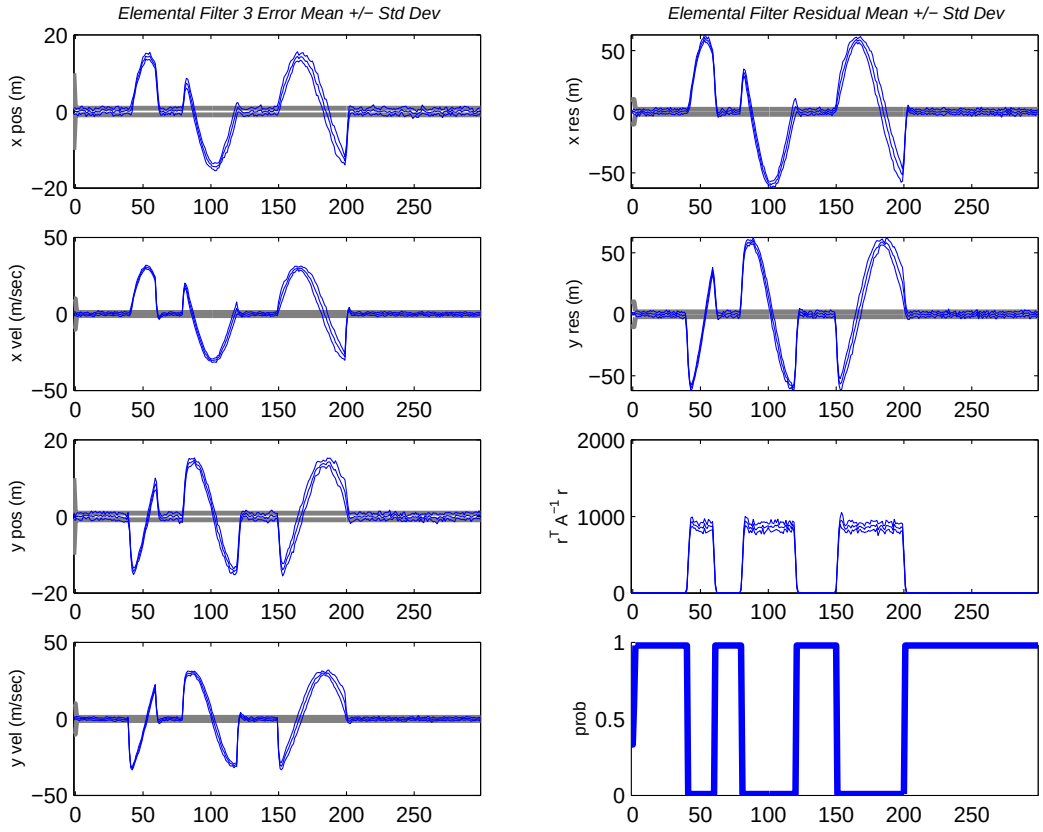

Figure 4.19: Filter 3 output. (Suite 5 vs. Scenario 5: MMAE)

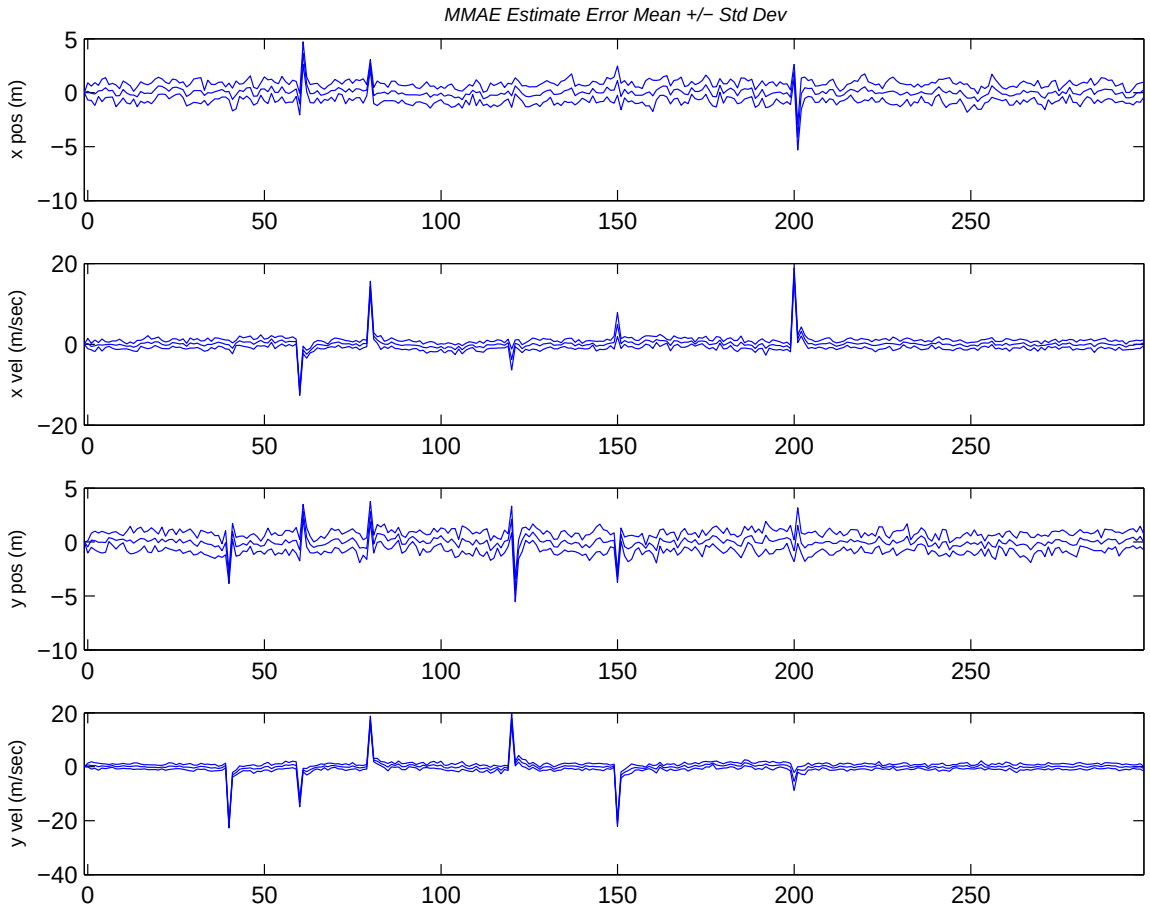

Figure 4.20: Blended estimate for a MMAE with a two, mirrored TPV filters and a CV filter running against a CV truth model executing three, +3-g TPV maneuvers over the course of the simulation. (Suite 5 vs. Scenario 5: MMAE)

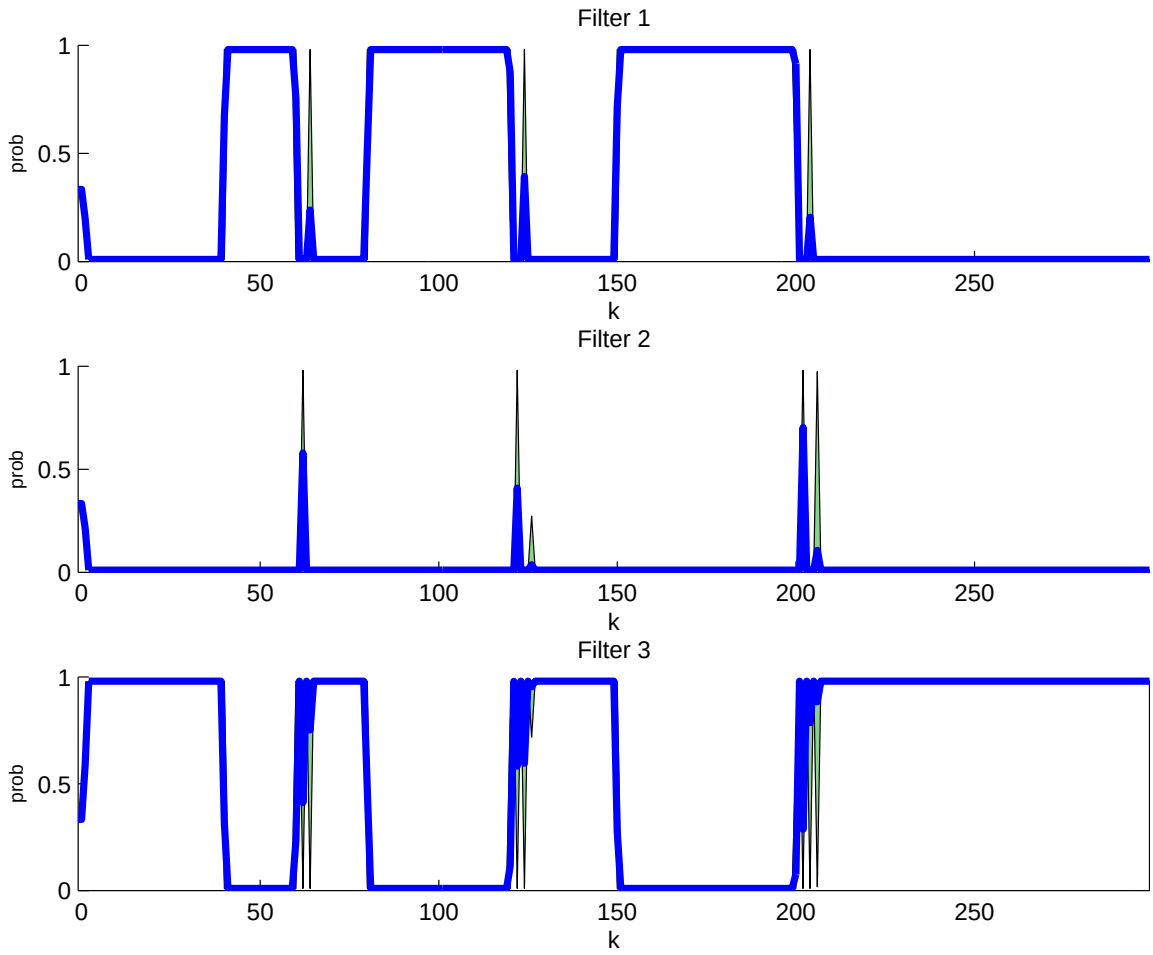

Figure 4.21: Probability flow for a system with a two, mirrored TPV filters and a CV filter running against a CV truth model executing three, +3-g TPV maneuvers over the course of the simulation. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

(and back again). In general, these effects can be mitigated by a higher-order filter that models jerk states in addition to position, velocity, and acceleration, but such a filter would need to be heavily tuned for specific transition effects. As a result, it would add significant computational burden to an MHT while providing benefit in only the most specialized cases.

Figures 4.21 and 4.22 shows the outputs for an IMM with a non-transition-favoring Markov matrix (see Appendix B, Section B.4) with filter Suite 5 running against Scenario 5. As expected, the probability flow transitions are less abrupt than for an MMAE (mainly at the beginning of the TPV maneuvers), and aside from these transient effects, the probability weight *does* stay with the appropriate filters during each phase of the truth trajectory. In general, however, the intermixing of estimates in an IMM reduces the distinguishability between elemental filters based on probability

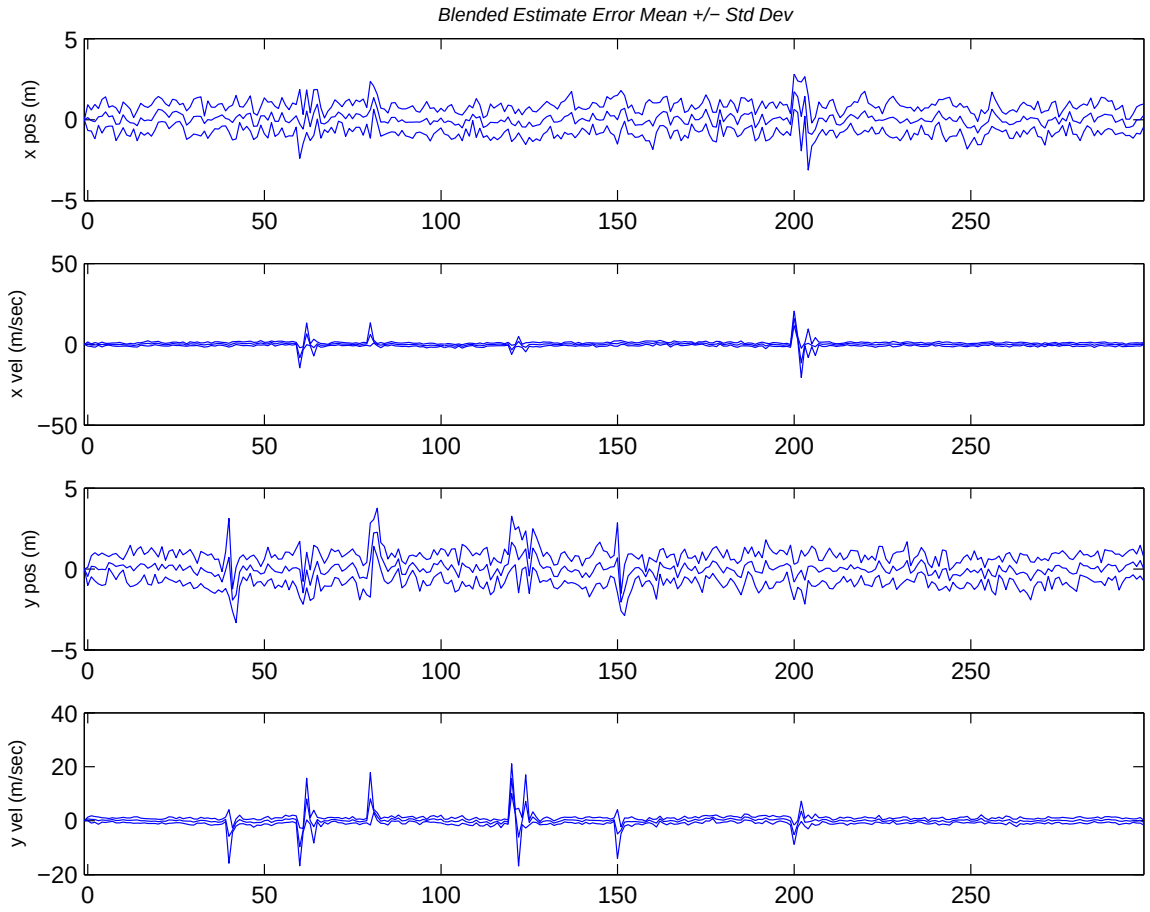

Figure 4.22: Blended estimate for a system with a two, mirrored TPV filters and a CV filter running against a CV truth model executing three, +3-g TPV maneuvers over the course of the simulation. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

weightings. The IMM's blended estimate, except for differences in plot scaling, exhibits similar performance to the MMAE for this configuration (compare Figs. 4.20 and 4.22).

Filter Suite 6 replaces Suite 5's CV filter 3 with a benign FOGMA model. An MMAE running filter Suite 6 against truth Scenario 8 has outputs depicted in Figs. 4.23 through 4.26. Notice, except for the elemental filter probability, the benign FOGMA filter's output (Fig. 4.25) is essentially identical to the single-filter case with the benign FOGMA shown in Fig. 4.15. Notice the +3-g TPV filter correctly takes most of the probability weight during the TPV maneuvers at $(k = 40, \ldots, 60)$, $(k = 80, \ldots, 120)$, and $(k = 150, \ldots, 200)$, and the benign FOGMA correctly takes most of the weight otherwise. Note that the shaded regions in the probability plots represent the maximum and minimum excursions of the probability weights seen on any of the 10 Monte Carlo trials[3]. The blended estimate is a marked improvement over the estimate that could be provided by the benign FOGMA alone. Its errors are almost entirely zero-mean, and there is little cross-correlation between the blended estimate errors and the truth model TPV maneuvers.

[3]Appendix A, Section A.1, gives a more detailed explanation of the probability flow plots and explains the motivation for this method of representation

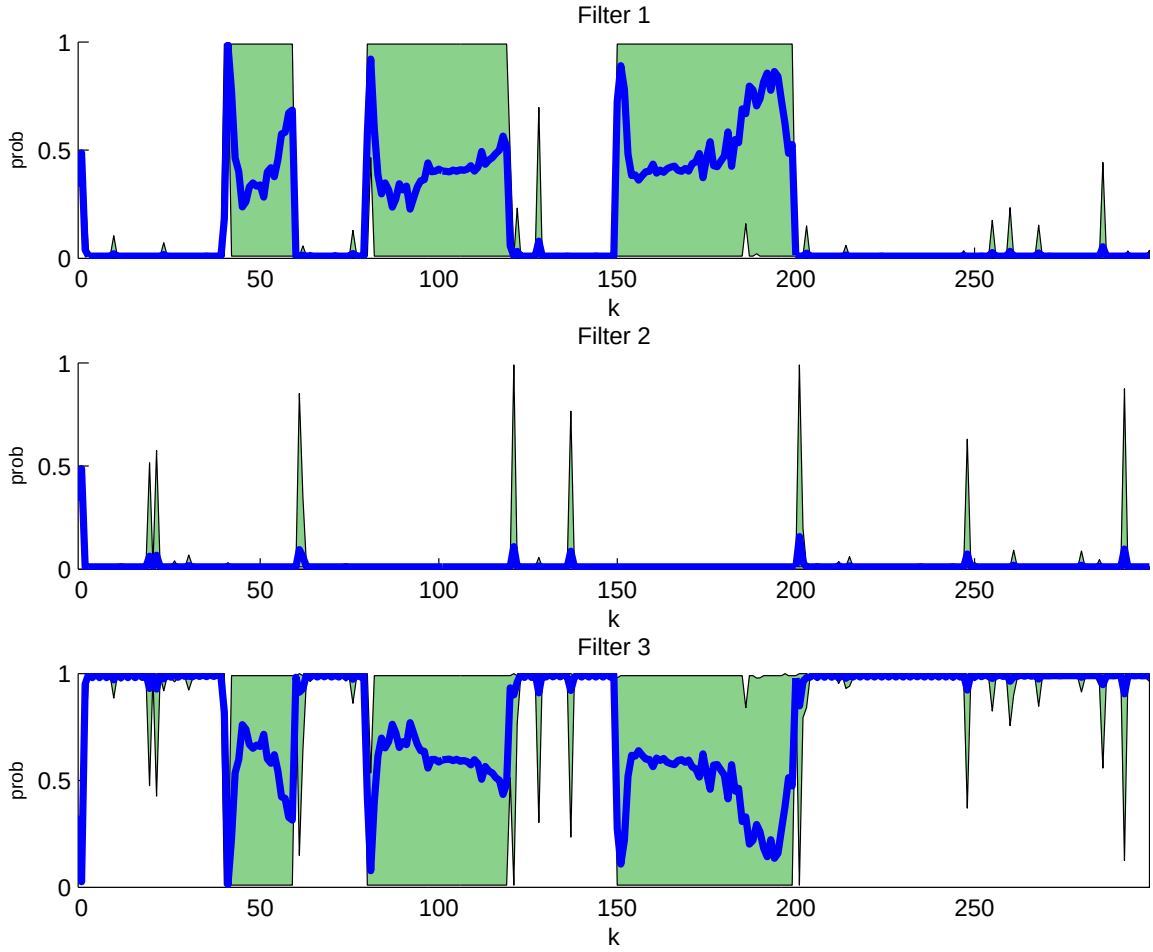

Figure 4.23: Probability flow for a system with a set of mirrored TPV filters and a benign FOGMA filter. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: MMAE)

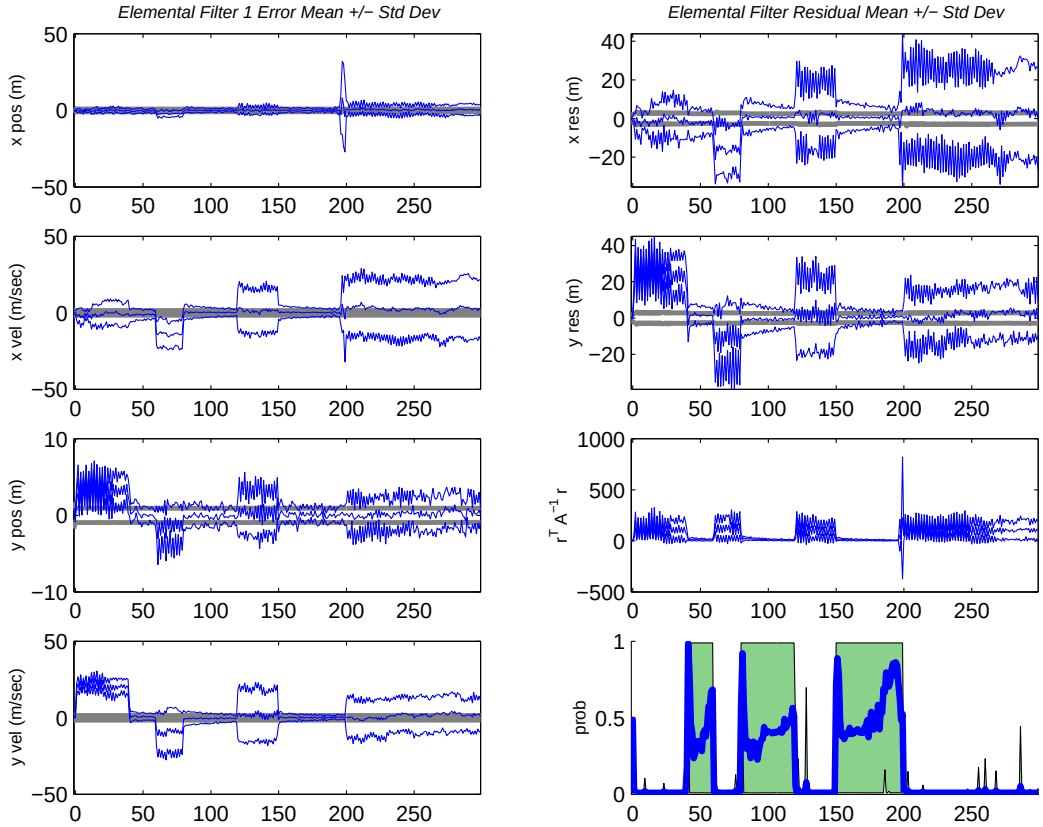

Figure 4.24: The +3-g TPV filter output in a system containing an additional set of mirrored TPV filters. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: MMAE)

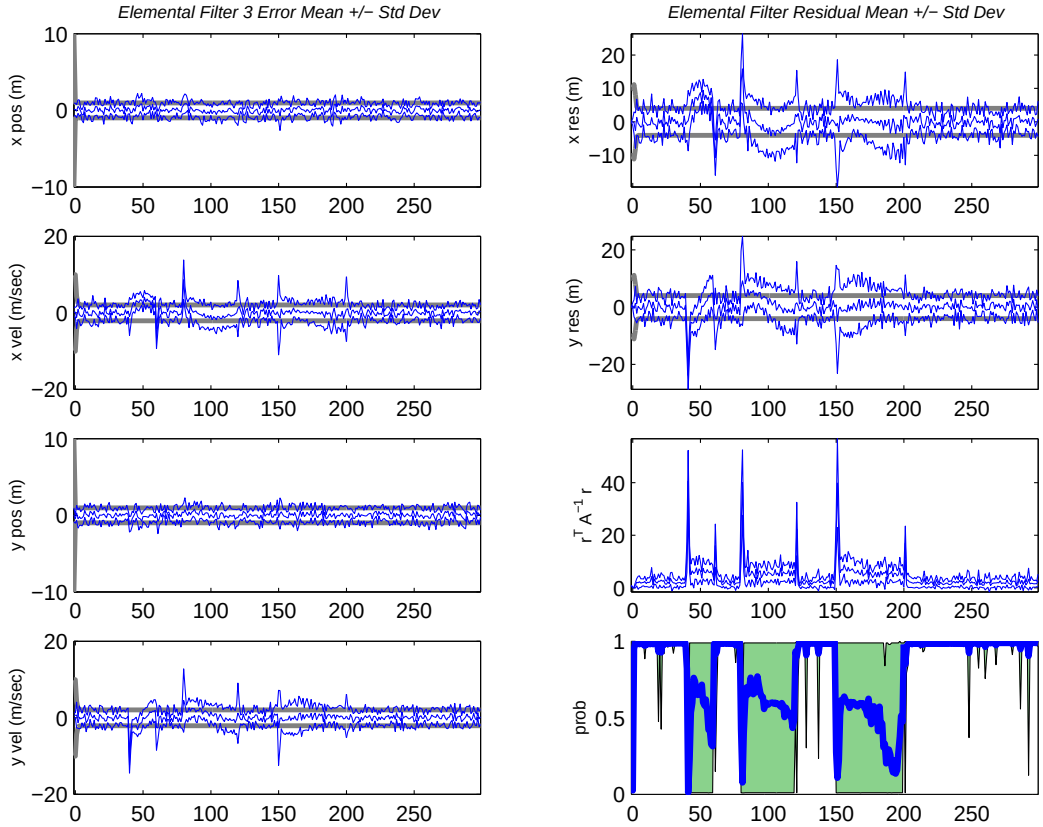

Figure 4.25: The benign FOGMA filter output in a system containing an additional set of mirrored TPV filters. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: MMAE)

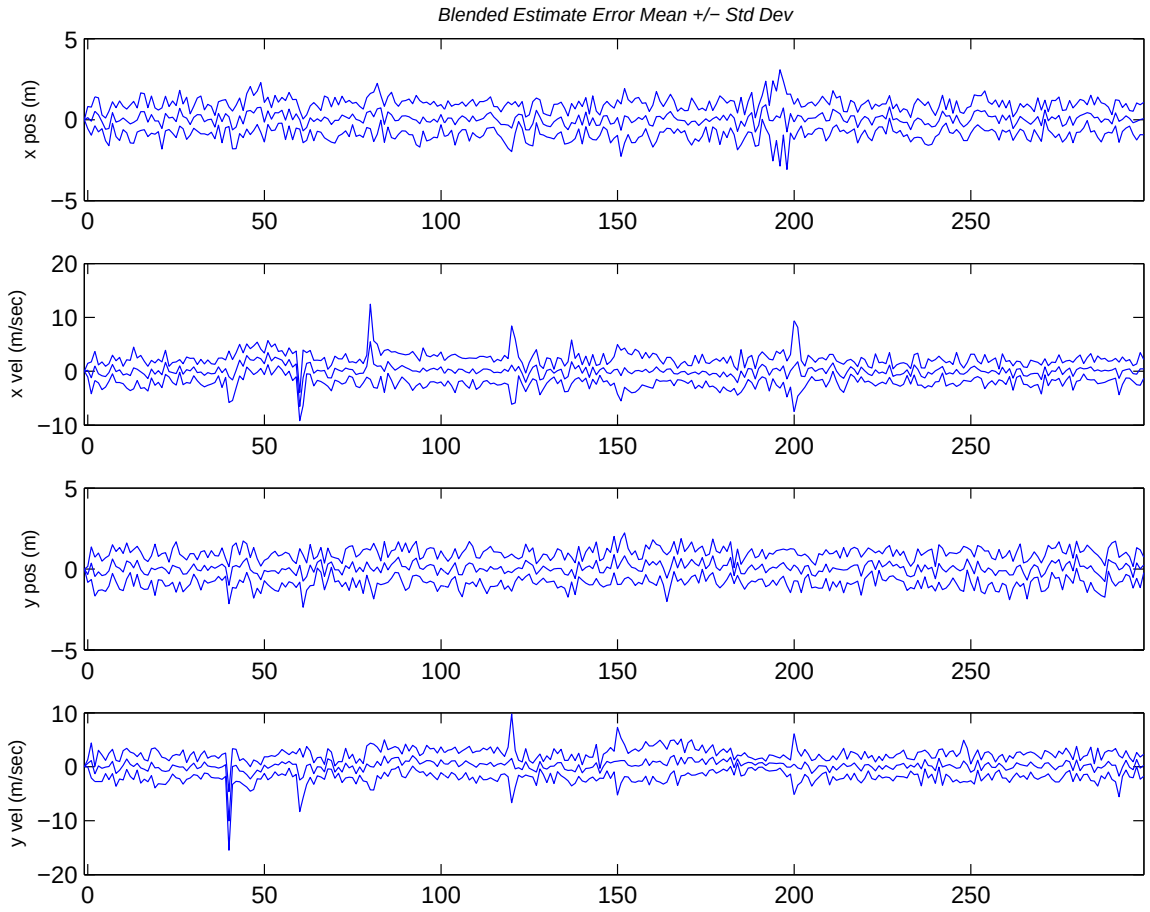

Figure 4.26: Blended estimate for a system with a set of mirrored TPV filters and a benign FOGMA filter. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: MMAE)

Figures 4.27 through 4.30 show the results from Suite 6 running against Scenario 8 using an IMM with a Non-Transition-Favoring Markov matrix. Notice again, except for the elemental filter probability, the benign FOGMA filter's output (Fig. 4.29) is essentially identical to the single-filter case with the benign FOGMA shown in Fig. 4.15. The blended estimate output is virtually indistinguishable from that of the MMAE, and like the case of the MMAE, exhibits mostly zero-mean error with little cross-correlation with the TPV maneuvers

In the outputs shown, probability flows appear slightly diluted compared to the flows seen when Suite 5 runs against Scenario 5 (see Fig. 4.16 or 4.21), but overall, the +3-g TPV filter was most heavily weighted during the +3-g TPV maneuvers. Unlike the case of Suite 5 running against Scenario 5, which involved only CV models, Suite 6 does not have a filter that exactly matches each phase of the truth trajectory. Suite 6 has a single, 6-state FOGMA with two, mirrored, 4-state TPV filters. During benign (non-maneuvering) phases, the FOGMA truth model in force precisely matches the model used as the basis for filter 3, the FOGMA filter. However, during TPV maneuver times, there is an underlying long-period, low-noise FOGMA also in force in the simulated truth environment, because software limitations did not allow the slightly more meaningful transition from benign FOGMA to a TPV maneuver executed on top of a *4-state CV model* (which is what the TPV filters are actually to model). The probability dilution is caused by the fact that the FOGMA filter, despite having a shorter time-constant and higher noise strength than the very benign FOGMA underneath the TPV truth maneuver, still represents FOGMA dynamics better than the 4-state TPV filters. However, the 4-state TPV filters match the TPV maneuvers better than the FOGMA filter does. In some respects, this is *exactly* what a multiple-model algorithm should be able to do (and does here): represent target dynamics which are poorly known, time-varying, and perhaps a *mixture* of deterministic and stochastic processes (i.e., a *mixture* of the hypotheses upon which the individual elemental filters are based).

A suite using TPV filters with long-period, low-noise FOGMA dynamics underneath was built and run against Scenario 8, but the performance difference was negligible and the 4-state TPV filters will reduce computational demands once MHT algorithms are tested. Modelling nuances aside, Suite 6 performed substantially better than the case with a single, benign FOGMA filter, and thus Suites 5 and 6 will be of particular interest in the MHT algorithms.

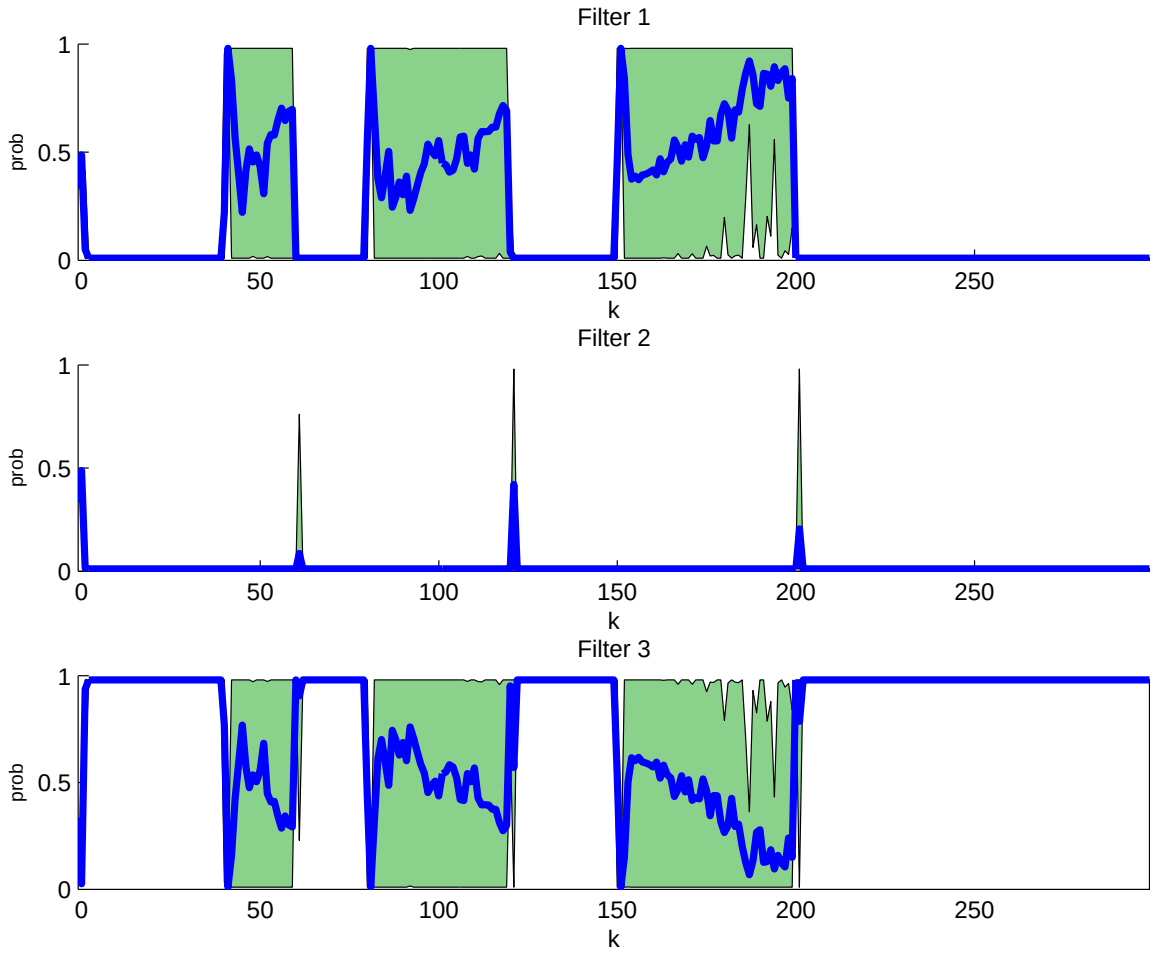

Figure 4.27: Probability flow for a system with a set of mirrored TPV filters and a benign FOGMA filter. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: IMM (Non-Trans. Favor. Markov))

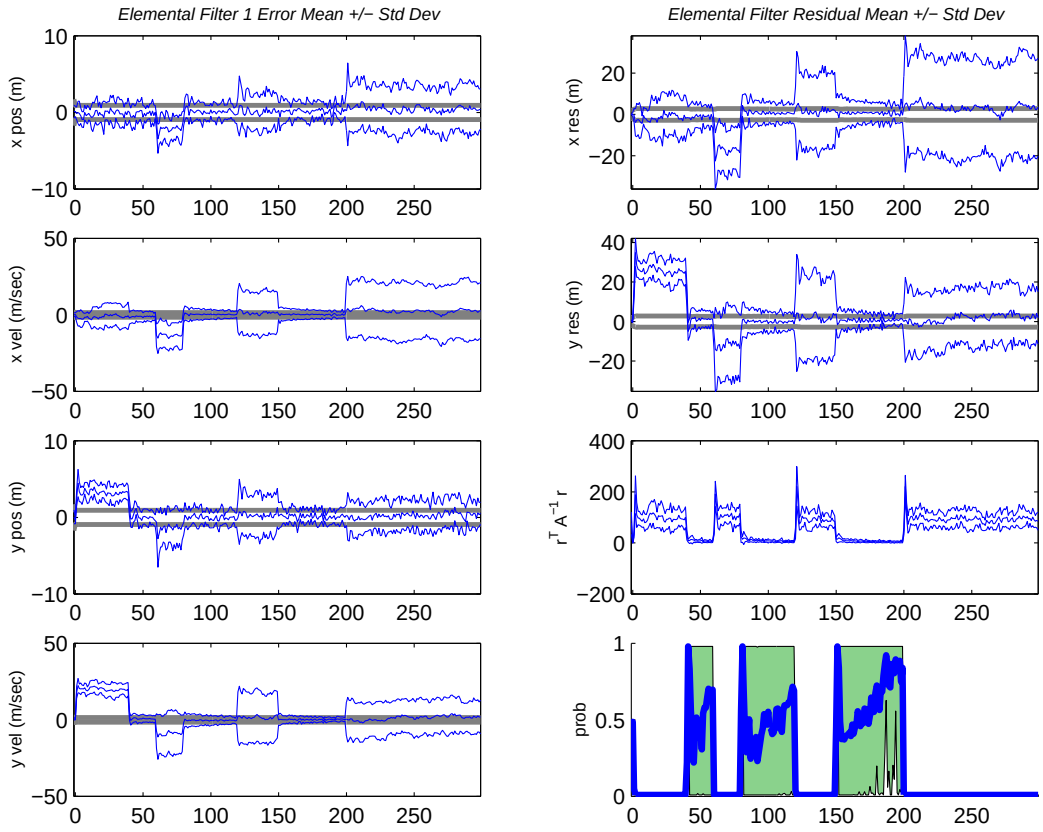

Figure 4.28: The +3-g TPV filter output in a system containing an additional set of mirrored TPV filters. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: IMM (Non-Trans. Favor. Markov))

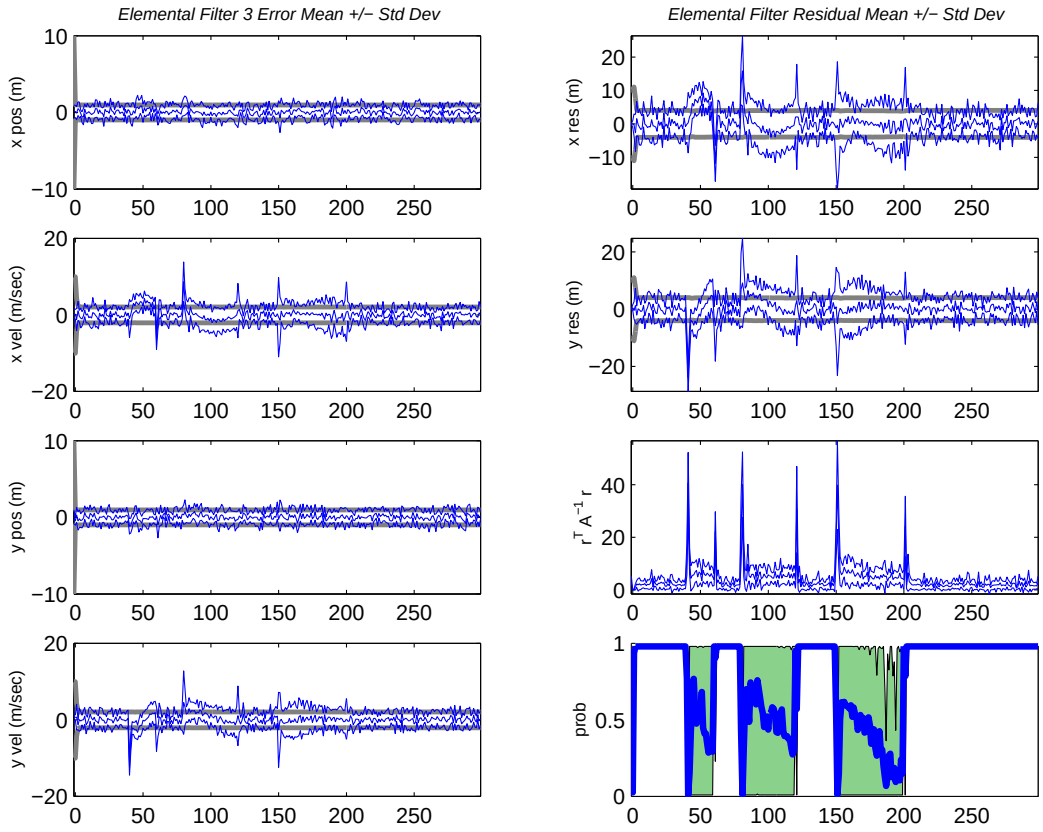

Figure 4.29: The benign FOGMA filter output in a system containing an additional set of mirrored TPV filters. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: IMM (Non-Trans. Favor. Markov))

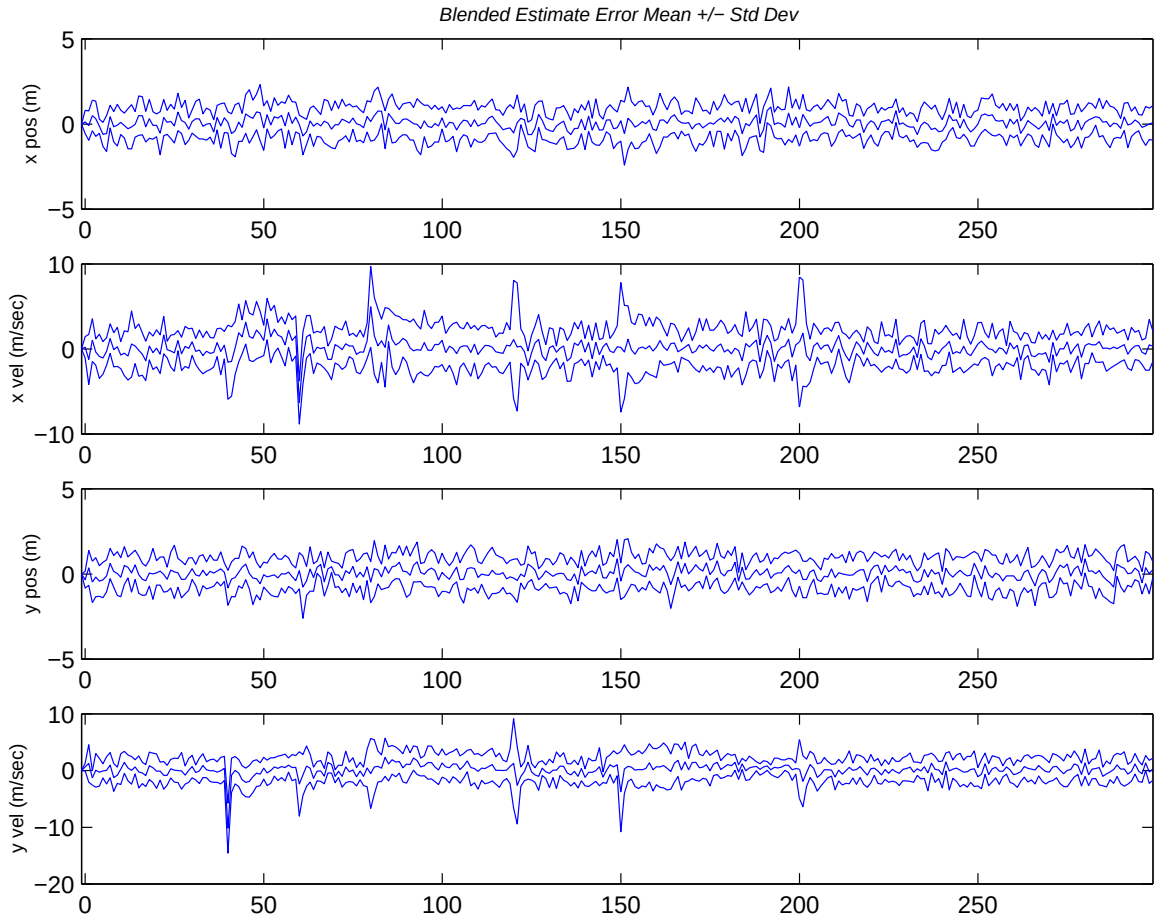

Figure 4.30: Blended estimate for a system with a set of mirrored TPV filters and a benign FOGMA filter. The truth trajectory includes three TPV maneuvers. (Suite 6 vs. Scenario 8: IMM (Non-Trans. Favor. Markov))

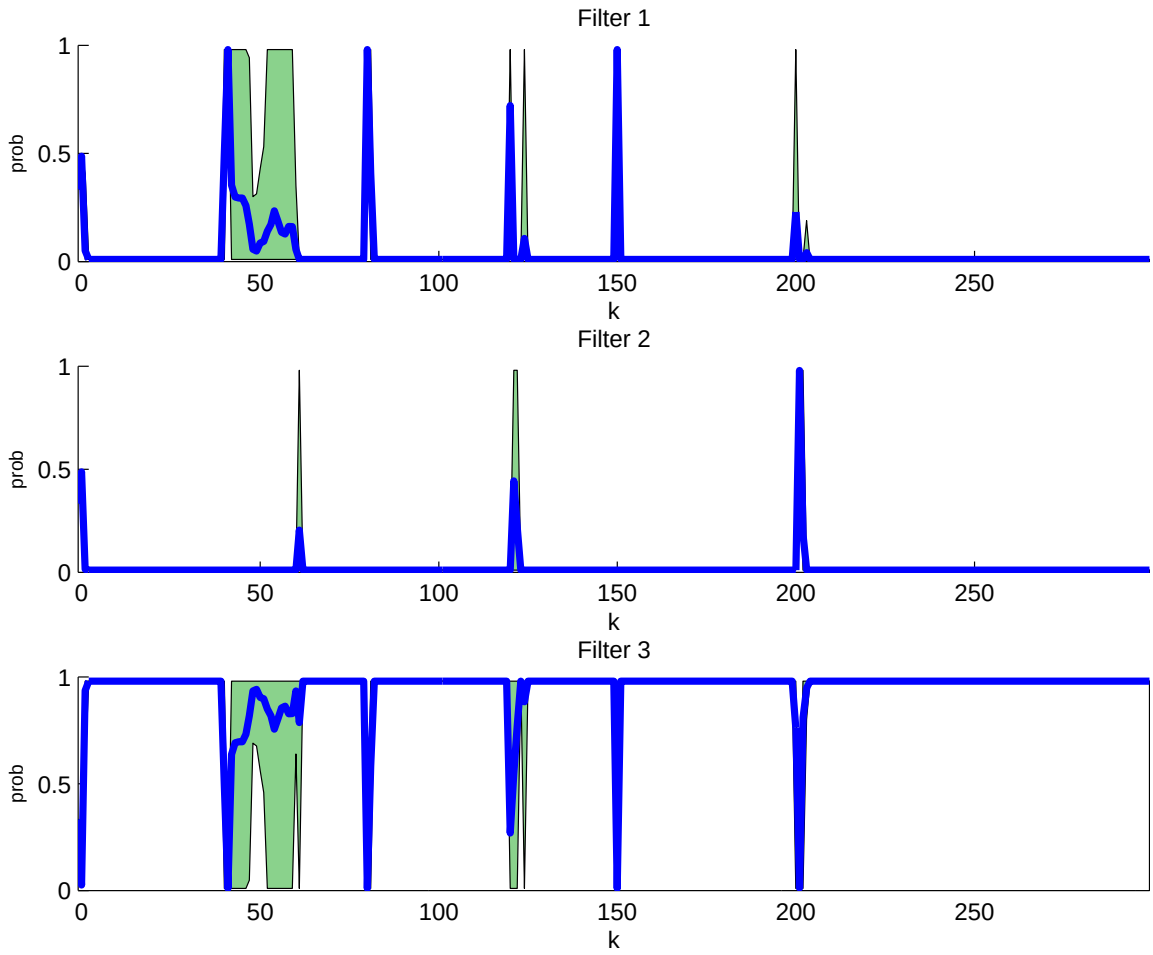

Figure 4.31: Probability flow for a system in which the TPV filters do not match the three, increasingly higher-g TPV maneuvers. (Suite 6 vs. Scenario 9: IMM (Non-Trans. Favor. Markov))

In light of how well the TPV filters estimate matching TPV maneuvers, there was question as to how well Suite 6 would handle TPV maneuvers with g-values that *did not match* those assumed by the filters themselves. Scenario 9 executed the same series of maneuvers as Scenario 8, but the TPV maneuvers were +4-g, +6-g, and +8-g instead of three, +3-g maneuvers. Figures 4.31 and 4.32 show results for an IMM with a transition-resistant Markov matrix. Notice in Fig. 4.31, the +3-g TPV filter catches the +4-g maneuver reasonably well, but the +6-g and +8-g maneuvers are handled by the FOGMA filter.

For the equivalent MMAE (see Figs. 4.33 and 4.34), none of the simulations progressed beyond a few samples past the +6-g maneuver. At that point, all filters became divergent, and the algorithm (as implemented in this research) could not recover. Divergence bounds were doubled from 40 to 80

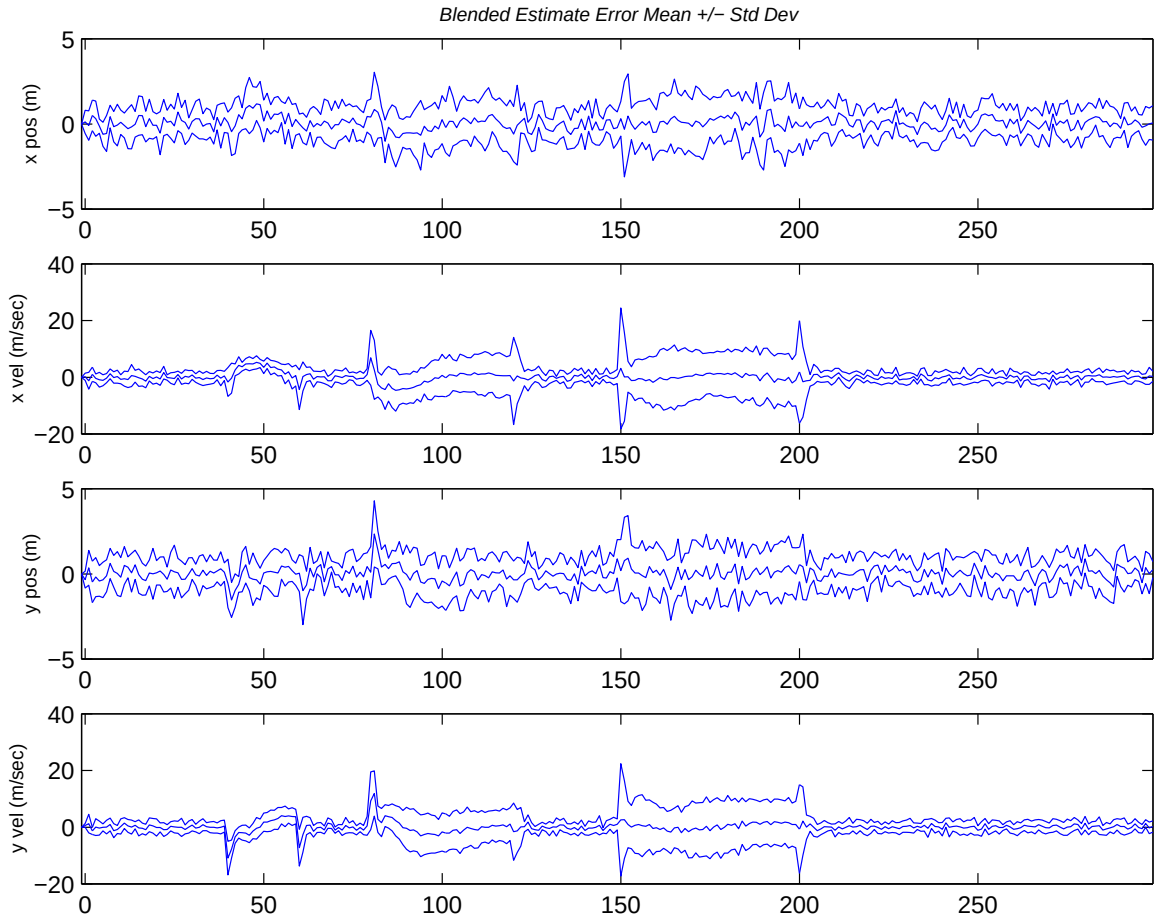

Figure 4.32: Probability flow for a system in which the TPV filters do not match the three, increasingly higher-g TPV maneuvers. (Suite 6 vs. Scenario 9: IMM (Non-Trans. Favor. Markov))

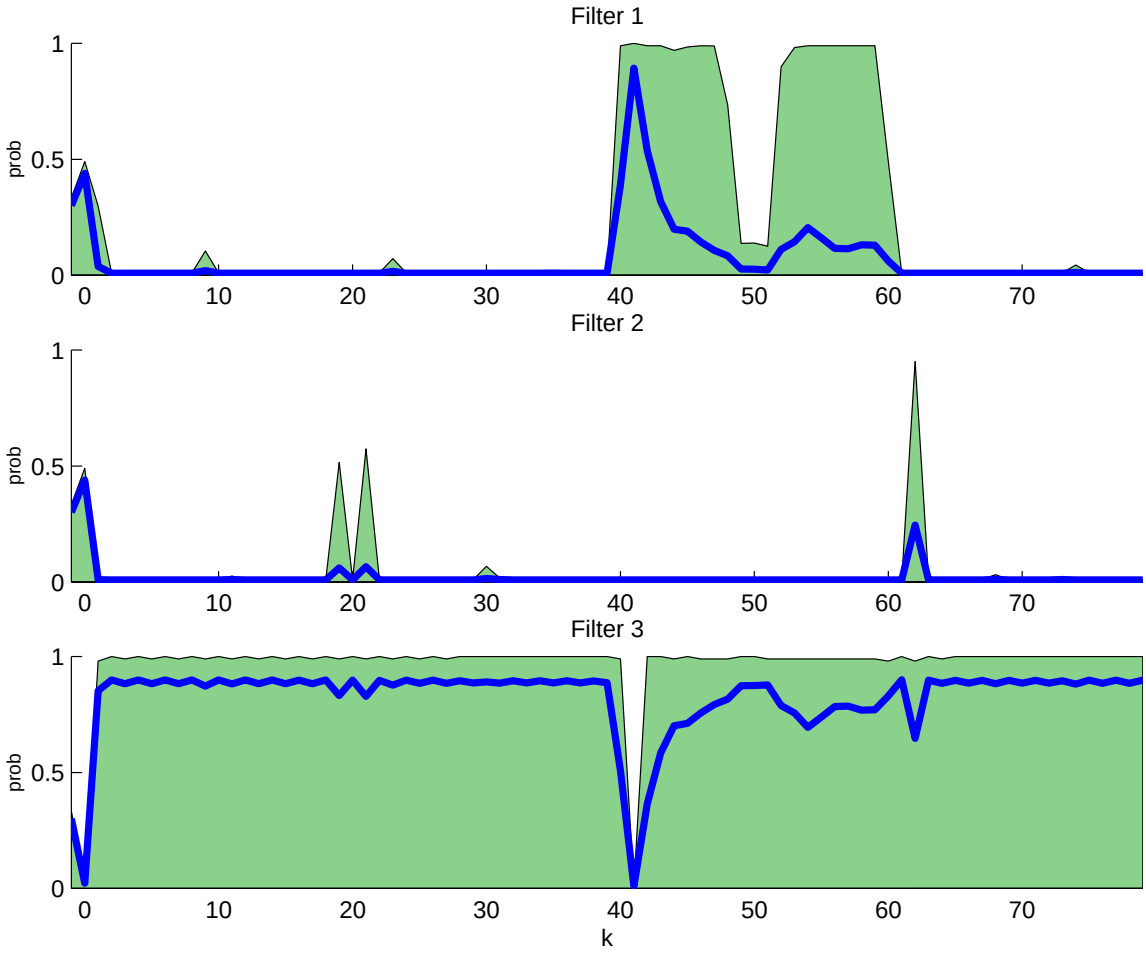

Figure 4.33: Probability flow for a system in which the TPV filters do not match the three, increasingly higher-g TPV maneuvers. The MMAE divergence bound was set to 40, which was standard for this research. (Suite 6 vs. Scenario 9: MMAE)

---

and the trials were run again, but the MMAE was still unable to progress beyond the +6-g maneuver. Finally, the divergence bound was increased to 2000, and the simulations ran to completion. However, the performance was essentially the same as the IMM's, which is expected since an MMAE without divergent filter resets is fundamentally identical to an IMM with an identity Markov matrix. An MMAE with a more realistic divergence bound (typically less than 200) will declare all of its filters to be divergent if none of them match truth well, and the tracker is left with no valid solution *unless* measures are taken to account for this special case of all filters being divergent. Such measures were not implemented in this research. Even though the IMM and the MMAE (with divergence bound set to 2000) carried through all maneuvers, the blended estimate is clearly not as good as could be obtained with a set of TPV filters well matched to each maneuver.

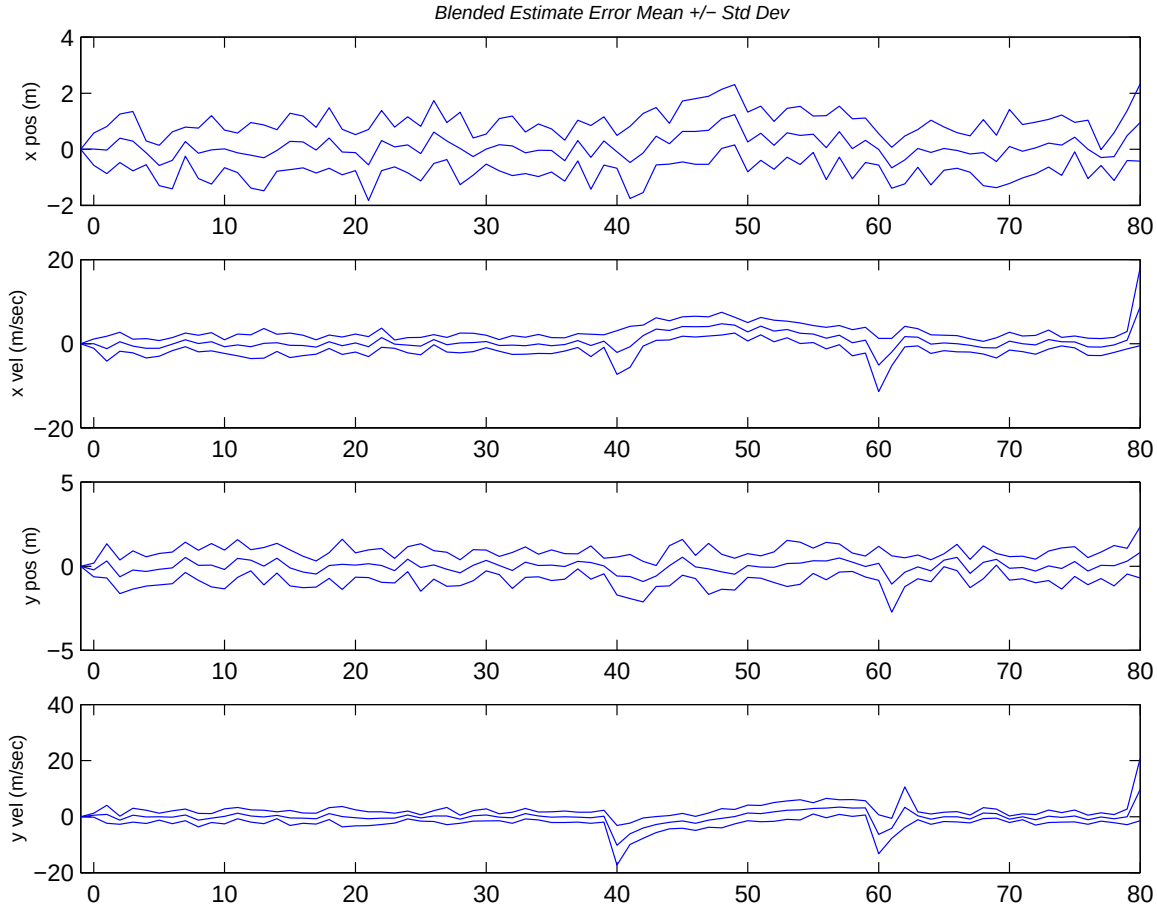

Figure 4.34: Probability flow for a system in which the TPV filters do not match the three, increasingly higher-g TPV maneuvers. The MMAE divergence bound was set to 40, which was standard for this research. (Suite 6 vs. Scenario 9: MMAE)

In early phases of testing the Kalman-filter-based configurations, when a benign FOGMA truth model was followed by a TPV maneuver, that TPV maneuver was executed on top of the benign FOGMA model (which was in force *for all time*). In other words, the only change to the truth model was the incorporation of the non-zero $\boldsymbol{B}_d(k)$ and $\boldsymbol{u}(k)$ terms. Scenario 6 exhibits this type of truth model progression. Contrast this with Scenario 8, in which the TPV maneuvers were executed over a FOGMA model with substantially longer period and lower noise than the "benign" phases of flight. Scenario 8 is arguably more physically realizable, because an airborne vehicle may meander in the measurement space during benign phases when it is moving along some unknown (to the sensor) but predetermined (from the perspective of the target vehicle's navigation system) flight plan towards some unknown (to the sensor) objective. However, during jinking maneuvers, this meandering is lessened, to an extent, and the deterministic TPV maneuver should probably be dominant. Scenario 6 assumes the meandering, benign FOGMA is still in effect during TPV jinking maneuvers, *but* the TPV maneuvers still dominate in magnitude. Scenario 8 attempts to reduce the effect of the FOGMA model so the dynamics essentially represent a constant velocity model. Again, software limitations prevented the most logical truth model progression: 6-state benign FOGMA to 4-state CV with TPV maneuver and back to 6-state FOGMA.

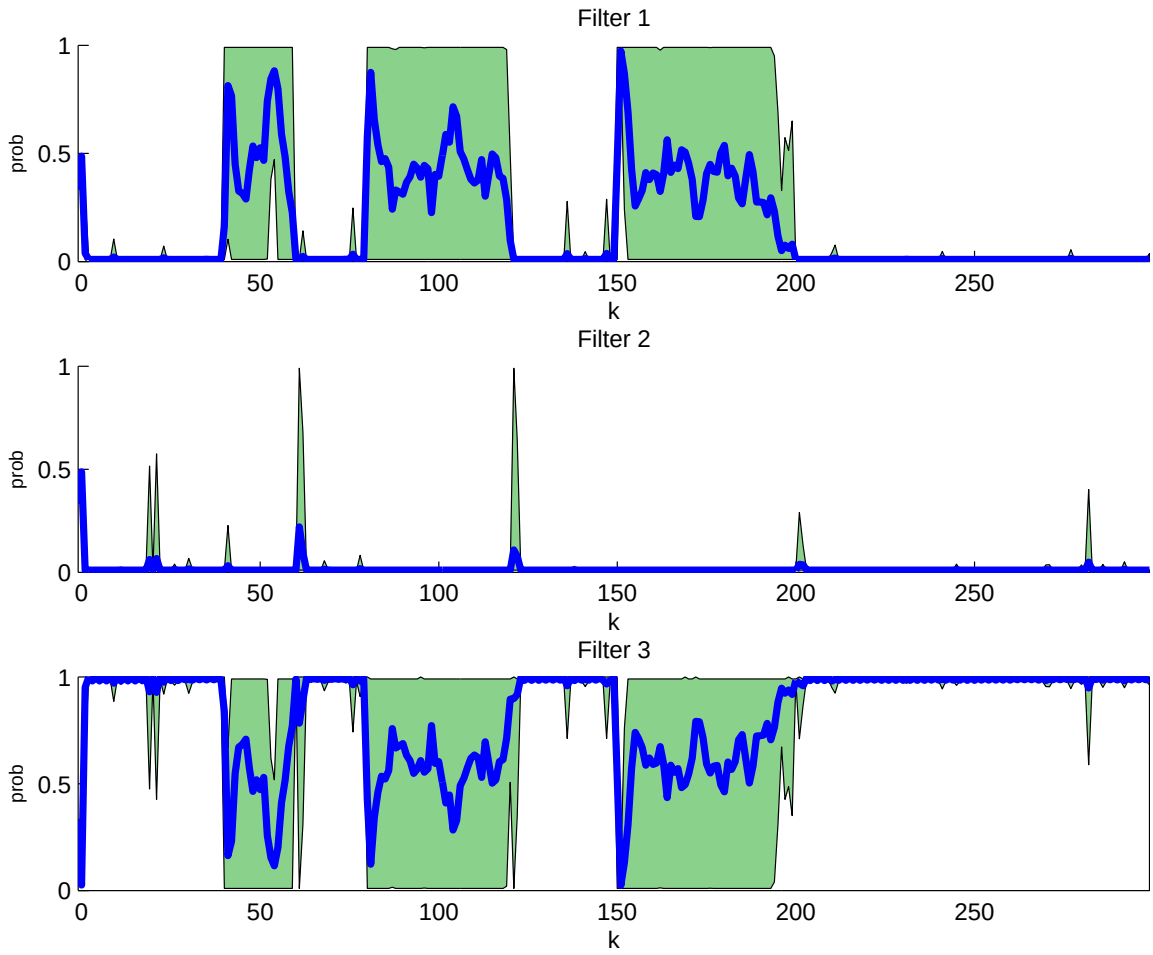

Figure 4.35: Probability flow (Suite 6 vs. Scenario 6: MMAE)

Having made the distinction between Scenarios 8 and 6, performance of Suite 6 against Scenario 6 was quite good. Figures 4.35 and 4.36 show the results for an MMAE. Figures 4.37 and 4.38 show the results for an IMM with a transition-resistant Markov matrix. The difference in performance between the MMAE and IMM is almost imperceptable, and some conclusions about relative performance will be drawn later in this section. Ultimately, both configurations recognized the TPV maneuvers quite well, despite the previously discussed tendency of the FOGMA filter to gain probability weight during the FOGMA-based TPV maneuvers.

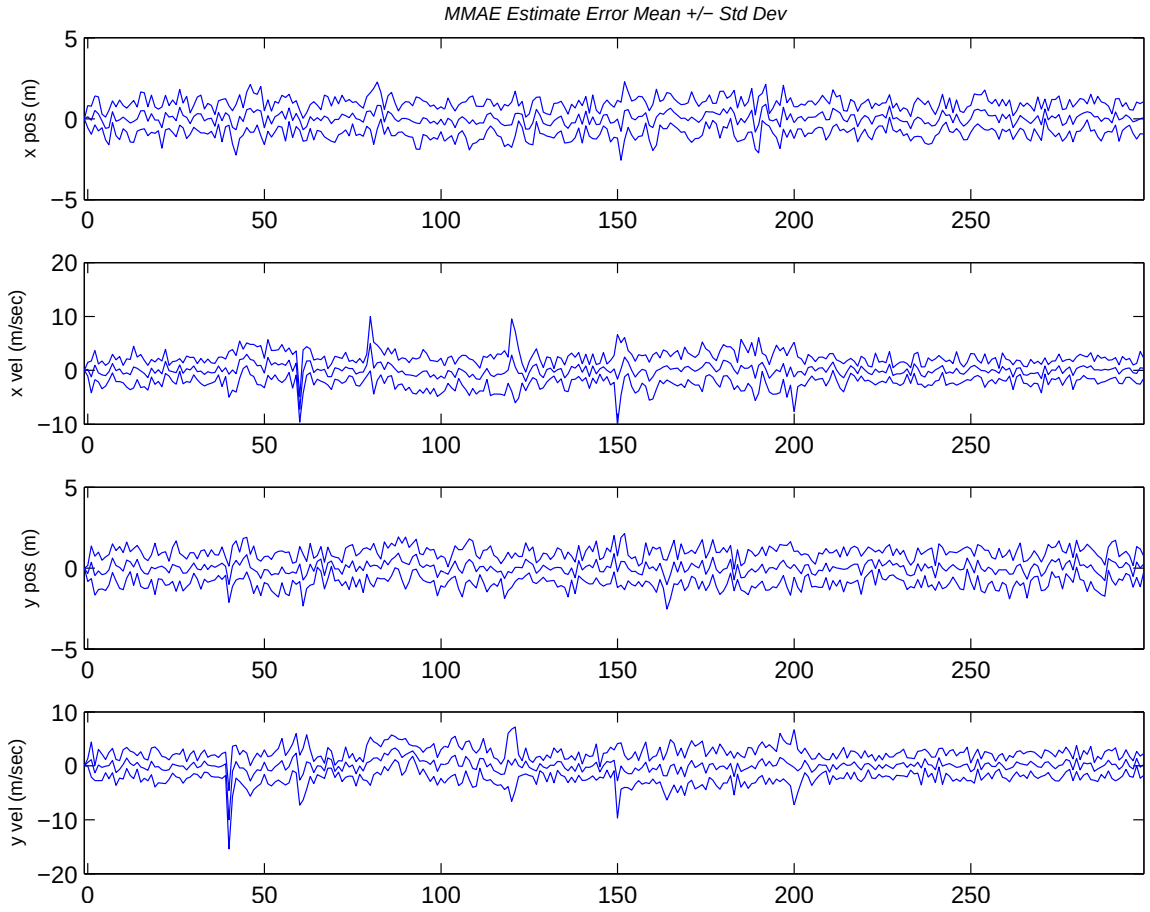

Figure 4.36: Blended estimate. (Suite 6 vs. Scenario 6: MMAE)

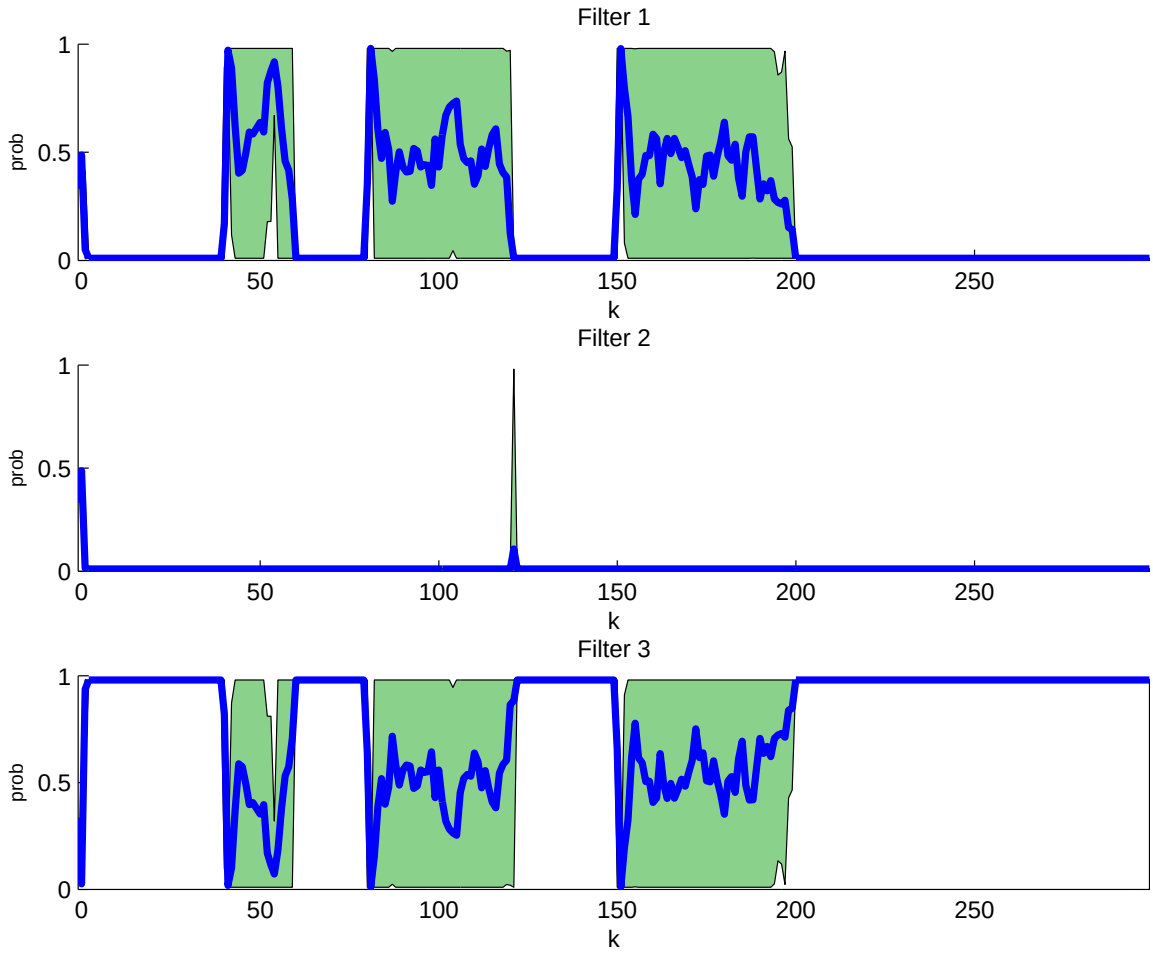

Figure 4.37: Probability flow. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

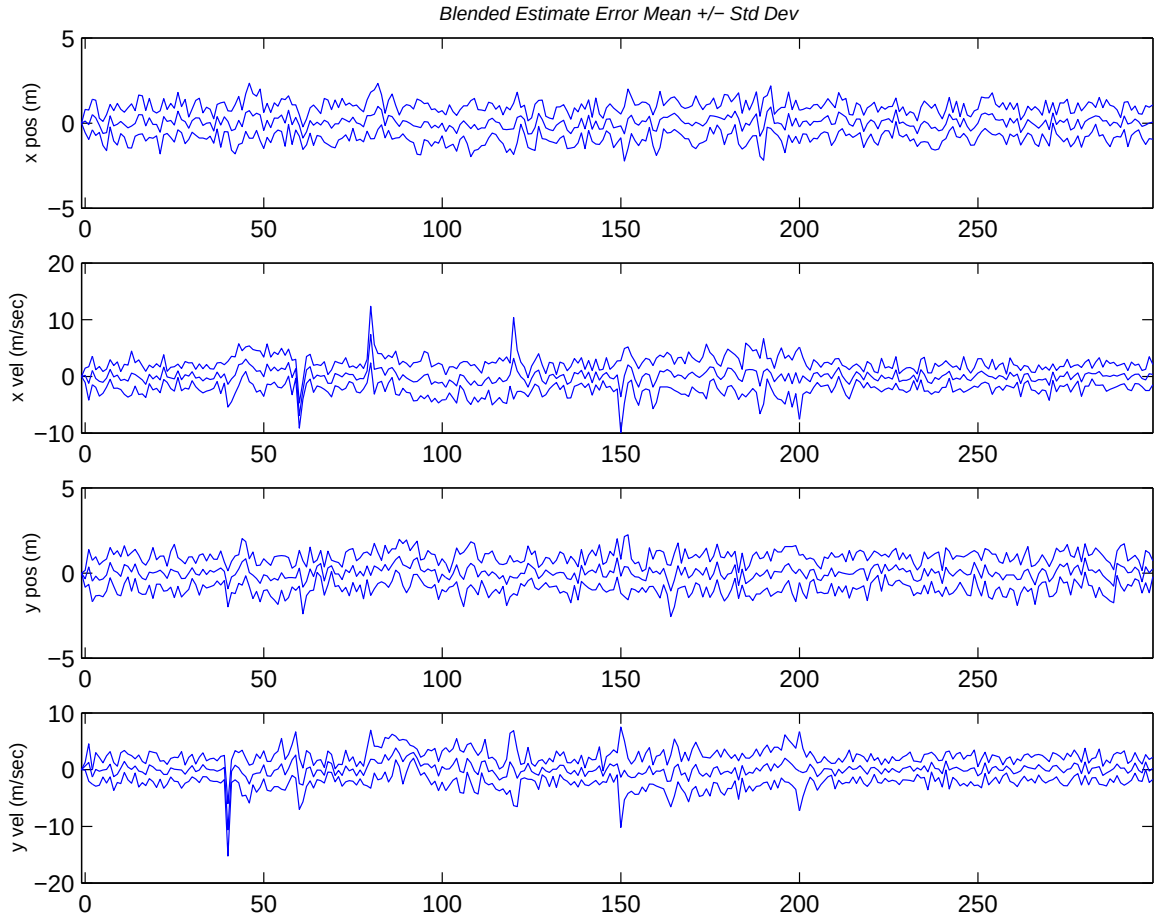

Figure 4.38: Blended estimate. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

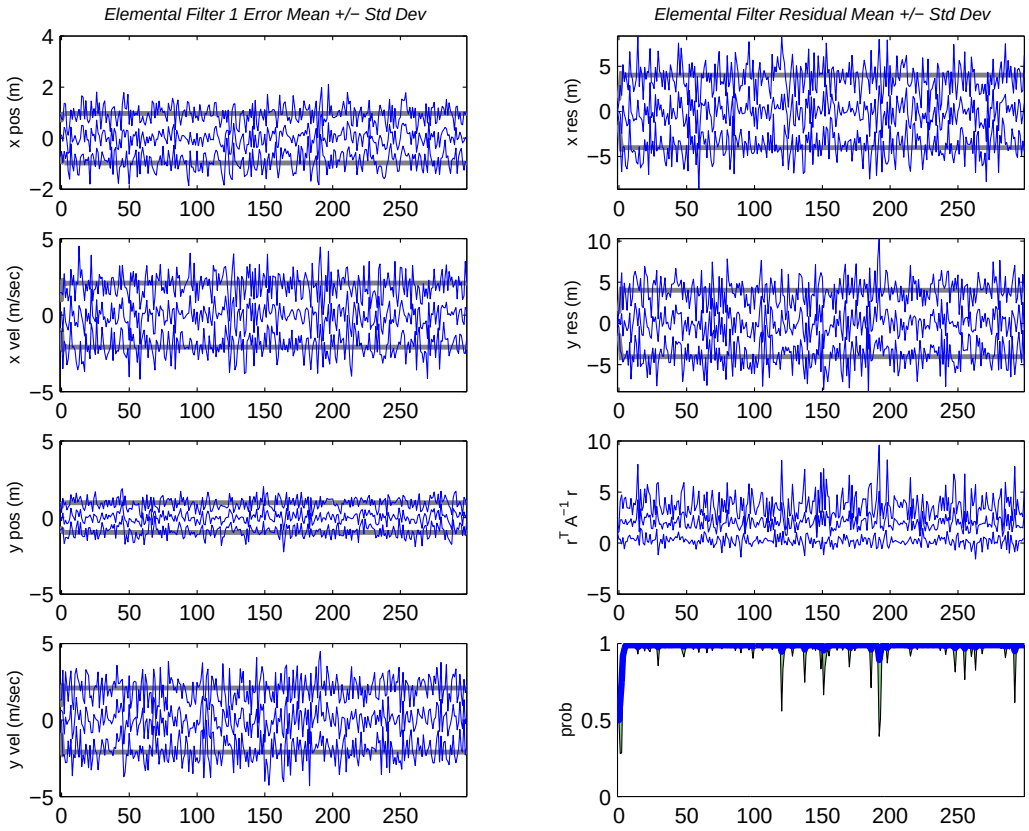

Figure 4.39: Filter 1 matches truth for all time. (Suite 4 vs. Scenario 2: IMM (Identity Markov))

4.1.7 Multiple-Model Configurations Against a Time-Invariant Truth Model. For completeness, some of the multiple-model configurations were tested against benign, time-invariant truth models to see how the multiple model architecture might adversely affect state estimate error when the truth dynamics are constant and known. These configurations will not receive a great deal of discussion in the MHT sections, because they do not add a tremendous amount of value to the discussion of algorithm performance. Nonetheless, since the Kalman-filter-based algorithms execute Monte Carlo simulations very quickly, it is worth taking a moment to consider any *negatives* to using a multiple-model approach.

Filter Suite 4, with two FOGMA filters, was run as an IMM with an Identity Markov matrix against Scenario 2, a time-invariant benign FOGMA truth model. Filter 2 matches truth for all time, and that filter's output is shown in Fig. 4.39. The blended estimate is shown in Fig. 4.40. Compared to Fig. 4.2, this output is essentially identical because the appropriate filter maintains virtually all of the probability weight for all time.

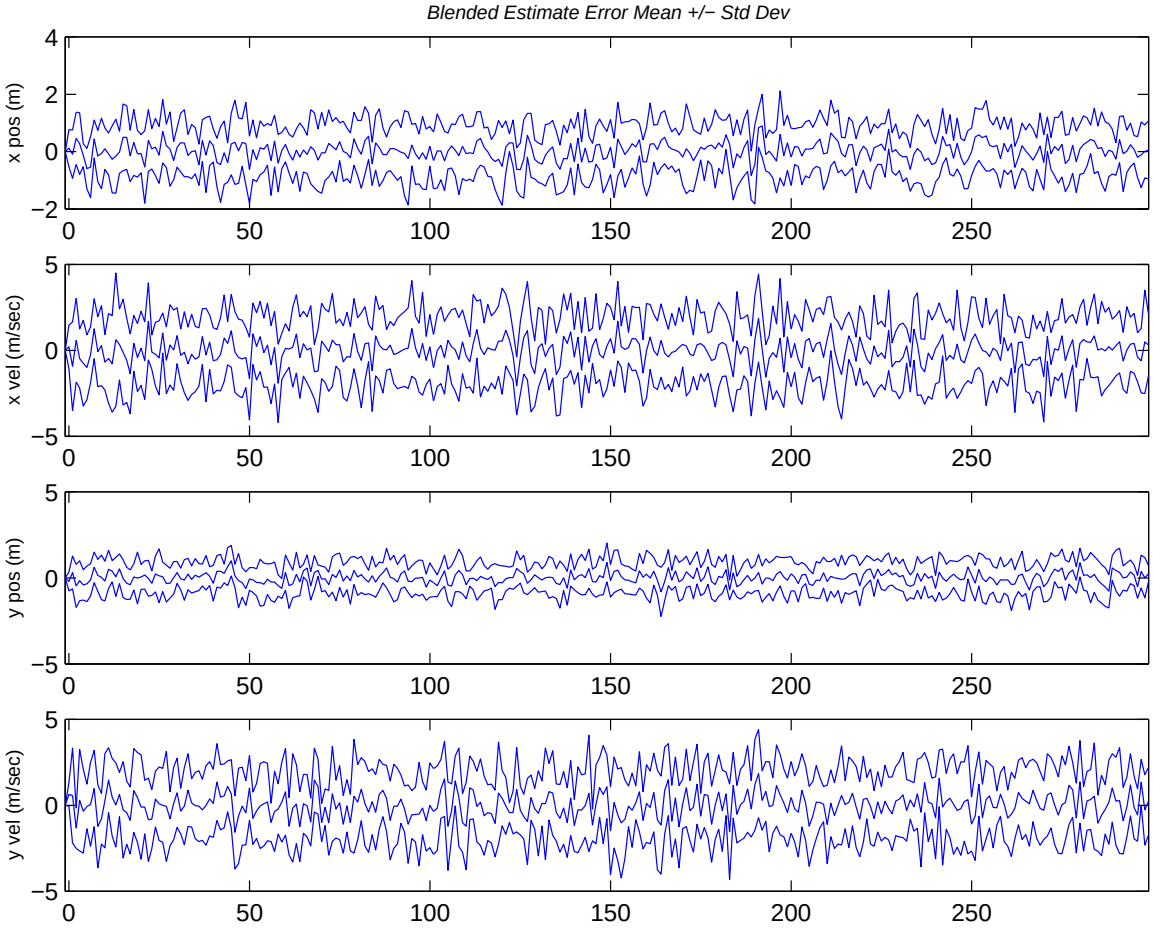

Figure 4.40: Blended Estimate. (Suite 4 vs. Scenario 2: IMM (Identity Markov))

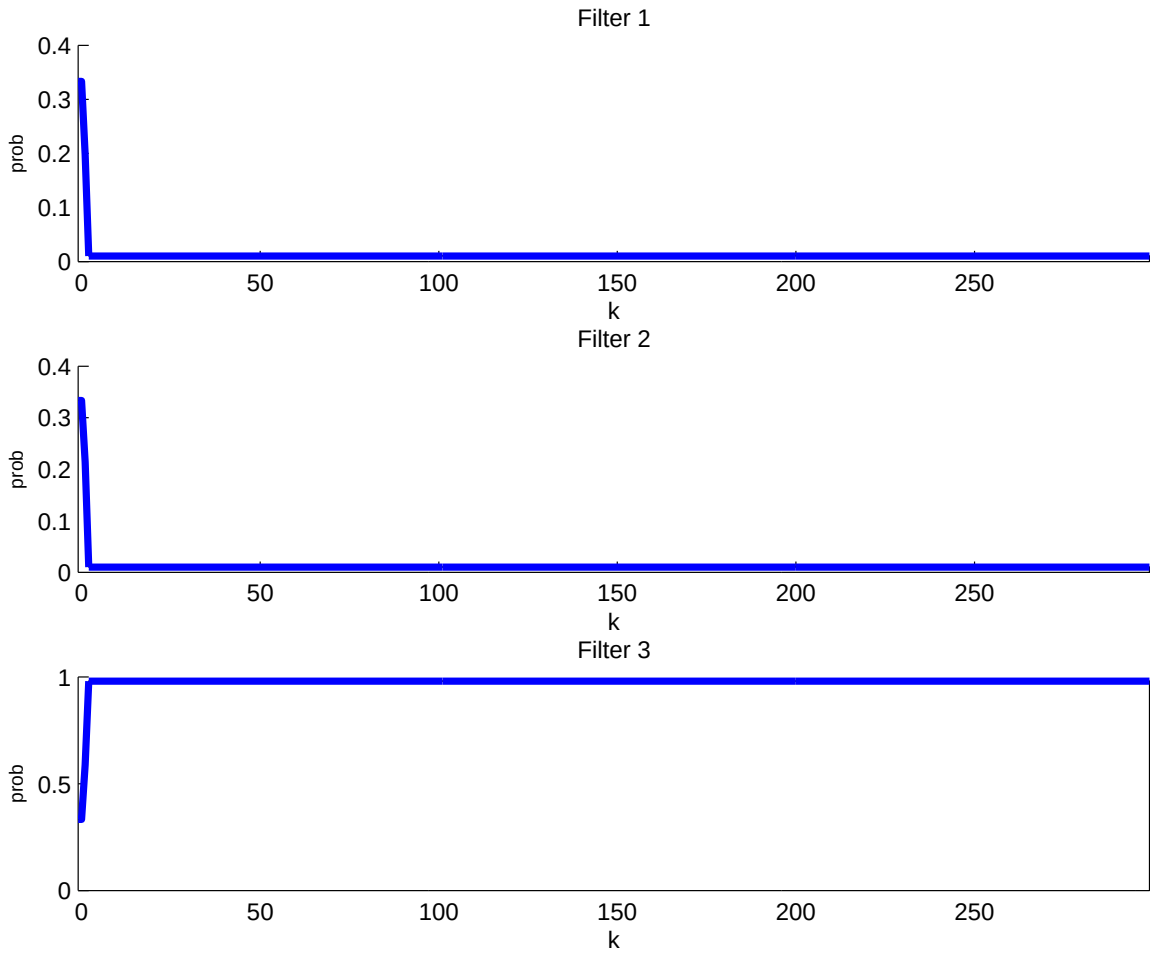

Figure 4.41: Probability flow for a three-filter system running against a time-invariant CV truth model. (Suite 5 vs. Scenario 1: IMM (Identity Markov))

Filter Suite 5, consisting of two, ±3-g TPV filters and one CV filter, was run as an IMM with an Identity Markov matrix against Scenario 1, a time-invariant CV truth model. Notice filter 3, which matches truth for all time, has all of the probability weight for all time (Fig. 4.41). Note that the blended estimate in Fig. 4.42 is virtually identical to the output in Fig. 4.1, the single-filter solution.

When truth dynamics are time-invariant and known well, there is no guarantee a multiple-model configuration will perform as well as a single-filter configuration matched to those truth dynamics. Because the multiple-model algorithms generally lower-bound modal probabilities, filters that do not match truth exactly will always create some contribution to the blended estimate even though the tracker would be better off having only the matched filter's estimate. Regardless, a

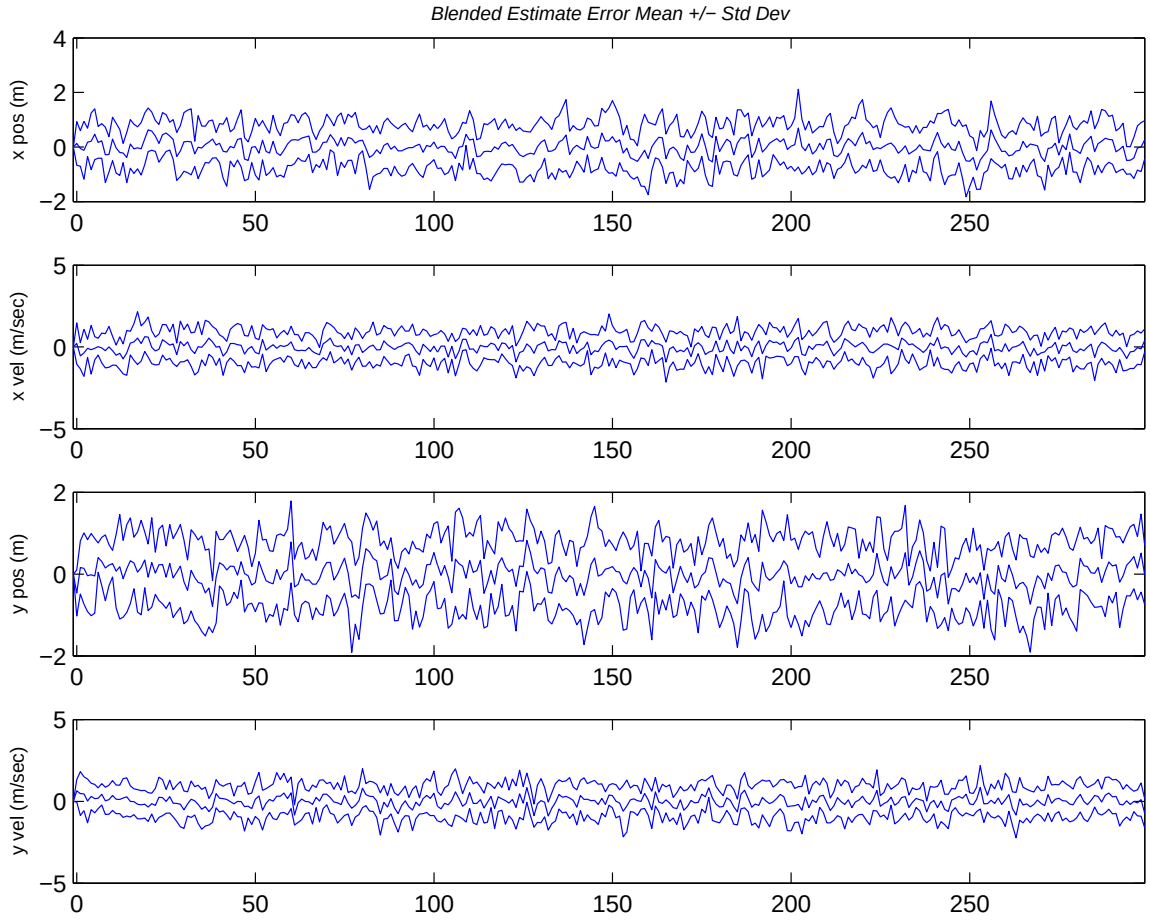

Figure 4.42: (Suite 5 vs. Scenario 1: IMM (Identity Markov))

well-tuned multiple-model configuration can provide a state estimate which is on par with, if not exactly the same, as that provided by the single, matched filter tracker. How close the multiple-model estimate is to the single-filter estimate will depend on what types of models are in the system, how the model parameter space is discretized, and how closely the truth model matches any one elemental filter in the filter bank.

4.1.8 IMM Configurations Using the Specific-Model-Favoring Markov Matrix. Although the Specific-Model-Favoring Markov matrix may appear to be clever at first glance, the performance of configurations using these matrices was decent but nonetheless poor compared to the other IMM configurations. Only one example will be addressed here, and that is filter Suite 4, with two FOGMA filters, running against truth Scenario 4 with its piecewise matched, time-varying FOGMA truth model. Figure 4.43 shows the blended estimate, which is not altogether unacceptable. However, as shown in Fig. 4.44, the probability weight stayed on filter 1 for all time even though filter 2 exactly matched the truth model from ($k = 100, \ldots, 199$). Looking at filter 1's summary output (Fig. 4.45), the residuals become quite large during the aggressive FOGMA maneuver, yet the Markov matrix is constantly forcing the probability weight towards this favored model. In light of the poor performance exhibited on several configurations, the Specific-Model-Favoring Markov matrix was not used in further simulation trials.

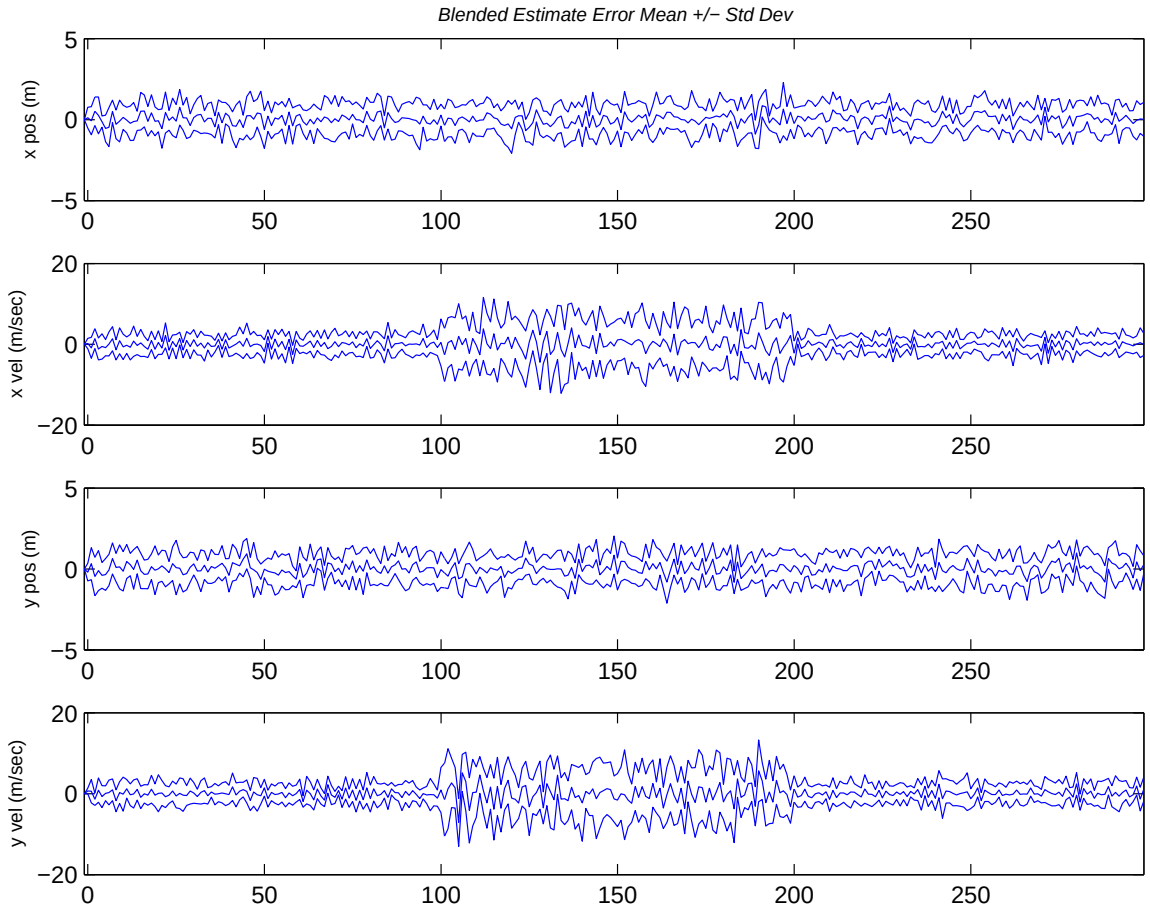

Figure 4.43: The blended output from the IMM using a Specific-Model-Favoring Markov matrix. (Suite 4 vs. Scenario 4: IMM (Specific-Model-Favor. Markov))

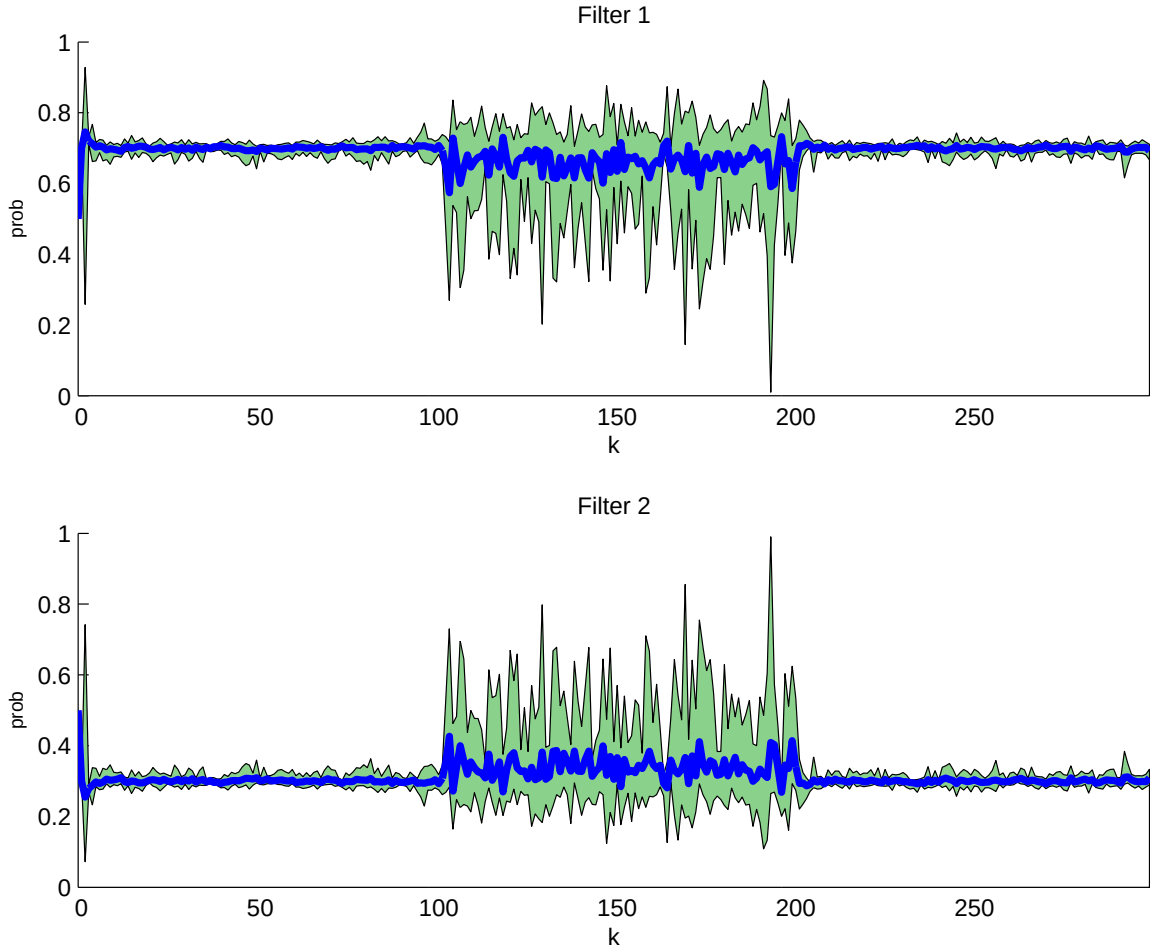

Figure 4.44: The probability flow from the IMM using a Specific-Model-Favoring Markov matrix. (Suite 4 vs. Scenario 4: IMM (Specific-Model-Favor. Markov))

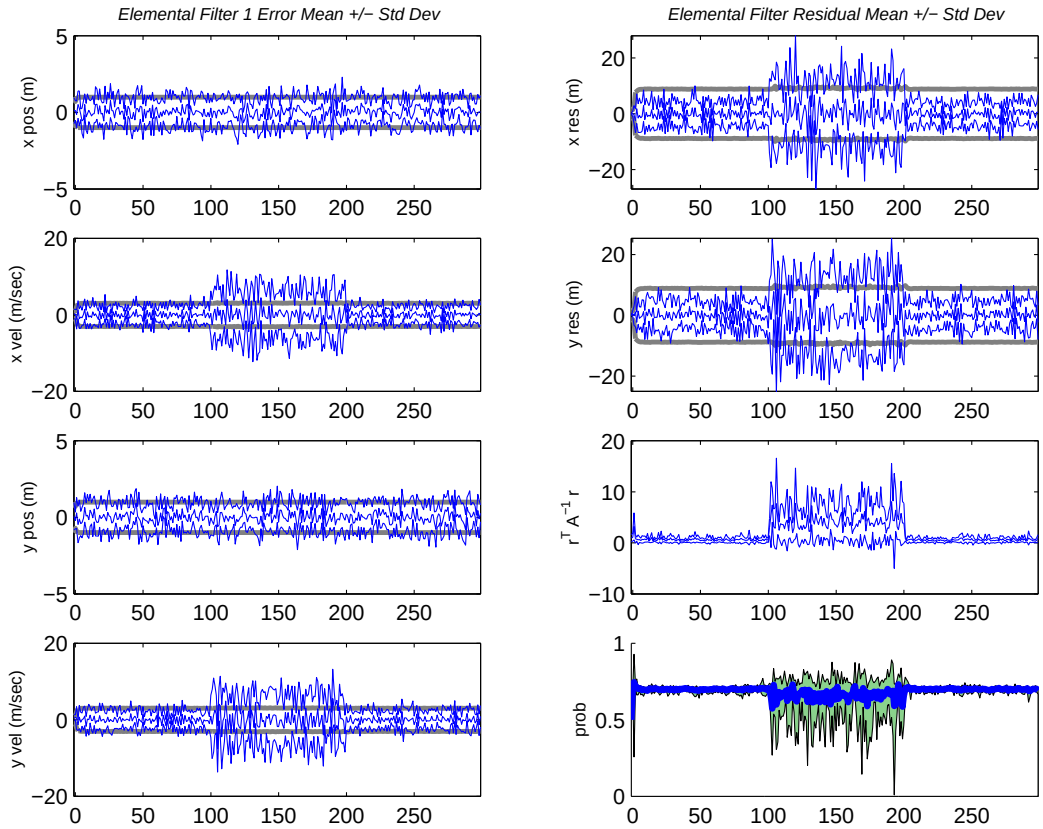

Figure 4.45: Filter 1 output from the IMM using a Specific-Model-Favoring Markov matrix. (Suite 4 vs. Scenario 4: IMM (Specific-Model-Favor. Markov))

4.1.9 Summary and Conclusions on Kalman Filter-Based Tracker Performance. As stated in the introduction, the Kalman-filter-based algorithms operating without measurement-association uncertainty provide a theoretical performance ceiling in the tracking problems of interest to this research. Therefore, the results described in the preceding sections are invaluable for making educated design choices in the construction of Williams-filter-based, MHT algorithms. Poorly performing Kalman-based suites will likely perform poorly when adapted to the MHT cases, while well-performing Kalman-based suites will likely perform respectably in the MHT case (in a general sense) even though specific behavior may differ substantially from the no-clutter cases.

Although the constant-velocity model has received a great deal of research attention in the past, it seems to handle anything beyond the most benign of maneuvers very poorly. A benign FOGMA seems to handle a constant velocity case quite well, but it provides added flexibility against a wider range of maneuver aggressiveness. Multiple-model, MHT configurations containing a CV filter will be examined in upcoming sections, but the FOGMA-based models are of more practical interest because of their versatility.

In cases with time-varying truth models involving no deterministic or strongly nonzero-mean maneuvers, the two-filter suites performed well, and their computational efficiency will be advantageous in the MHT algorithms. When deterministic, TPV maneuvers are present, few substitutes exist for a dedicated set of TPV filters. Unfortunately, as shown when Suite 6 was run against Scenario 9, TPV filters have a relatively narrow band of g-loadings in which they can recognize a truth model TPV maneuver. In an ideal situation, the parameter-space discretization for the TPV filter g-values would be designed to accommodate the spectrum of maneuvers expected by a theoretical target. This would be feasible in the Kalman-based algorithms, but the Williams-filter-based MHT algorithms would suffer severely from the computational burden of more than one set of TPV filters. For this research, concentration will be on a single, mirrored pair of 3-g TPV filters with the understanding that: 1) this represents a severe compromise, and 2) the choice of g-value for the two filters is dictated by the g-value executed in a truth TPV maneuver and is, therefore, somewhat artificial.

Performance differences between MMAE and IMM using the Kalman-filter-based algorithms are not significant at this point. As expected, when the truth model executes maneuvers that correspond precisely to a model in the filter bank, a well-tuned MMAE will tend to shift probability weight heavily towards the matching-dynamics filter. An IMM with anything other than an identity Markov matrix will force some degree of interaction among all filters in the system even when truth precisely matches a filter in the bank. Figure 4.21 provides the best example of this, because the corresponding MMAE's probability flow (Fig. 4.16), which has no noticeable extended transients

(i.e., the probability flow is a sharp step function), better illustrates what the truth model is actually doing. In theory, the IMM's interaction should make that filter bank more *agile* or better able to detect changes in truth dynamics after a sustained period of a given dynamics model. None of the IMM configurations in this research exhibited that quality, but only a limited number of filter suites and truth scenarios were tested. In all of the simulations without measurement association uncertainty, MMAE and IMM appeared to have virtually identical state estimate error performance even though probability flows were not necessarily similar.

Before proceeding any further, it is worth noting that the filter suites and truth scenarios used in this research are best cases. That is to say, in a real-world application, a truth model will not obey a predetermined and time-invariant dynamics model for some finite phase of its flight. So, unlike the assumptions made in this research, the model that best describes that target: 1) would be a continuous-time model, 2) would have continuously-changing dynamics parameters, and 3) would best be represented using more dynamics state variables than those used here. The implication for this research regarding the relative performance of IMM and MMAE is that both of these algorithms benefit from the finite truth model set and piecewise-time-invariant truth dynamics. However, even in a realtime, real-world tracking scenario, the *tracking filters themselves* will almost always be discrete-time equivalent systems of low-order (most likely with acceleration being the highest-order derivative) with piecewise-time-invariant dynamics parameters.

As will be noted in Chapter 5, a more accurate trial would be against some predefined set of trajectories produced by a trajectory generation routine. Such a routine could generate relatively high-order models with continuously changing parameters much more effectively than the rather limited simulation environment used for this research. Using these trajectories, the MMAE and IMM algorithms would truly be tracking a realistic target of poorly-known dynamics, and the resulting performance could be significantly different than that seen here. The effects of parameter space discretization and *ad hoc* design choices would probably be more evident, since the algorithms would be forced to adapt their finite model set to a target with a virtually continuous spectrum of truth dynamics models. The cases in which filter Suite 6 was run against the increasingly higher-g TPV maneuvers of truth Scenario 9 give an indication of the performance difference when filters and truth do not exactly match.

4.2 Algorithm Modifications in MHT Implementations

Chapter 3 discussed in detail the modifications necessary to adapt the Kalman-filter-based multiple-model algorithms to MHT trackers incorporating Williams hypothesis reduction filters. In the presence of measurement-association uncertainty, Gaussian approximation assumptions allow

the pseudo-states, pseudo-residuals, and pseudo-error covariances to take the place of the individual states, residuals, and error covariances present in the conventional MMAE and IMM. Furthermore, defining points of entry and exit into the Kalman filter algorithm clarified the steps necessary to accomplish a "black-box" replacement with a Williams filter.

Initial testing of the MHT algorithms highlighted a few deficiencies in the implementations derived from developments in Chapter 3. Subsection 4.2.1 will discuss the need to lower-bound the modal probabilities in both MMAE and IMM, even though the basic IMM algorithm does not call for lower-bounding. Subsection 4.2.2 will discuss the concept of measurement gate unions and the dramatic effect that has on MHT performance. Subsection 4.2.3 will discuss the motivation to lower-bound Gaussian mixture component probabilities within each filter in light of measurement gate unioning. And finally, Subsection 4.2.4 will discuss a method that addresses the need to restart divergent filters (in MMAE) or mix filter estimates (in IMM) when those filters are based upon Gaussian mixtures as opposed to a single Gaussian density, with state estimate and error covariance that completely define that density.

4.2.1 Lower-Bounding of Modal Probabilities. The essence of a Multiple Hypothesis Tracker is its ability to defer a decision about which *track* (or association history hypothesis) is indeed correct until some point in the near future. In a single-filter MHT, each hypothesis is propagated using the same dynamics model, and component probabilities have a tremendous influence on the choices made by the hypothesis reduction algorithm as it merges and prunes existing mixture components. The single dynamics model adds an element of simplicity to the problem, because the probability weights are assigned under the assumption *that all components are obeying the same dynamics* and, therefore, *components lying far from the center of the Gaussian cluster must be the result of some spurious measurement.*

In the multiple-model MHT algorithms used in this research, the component reductions occur inside each elemental filter, so all components propagate according to the same dynamics as in the single-filter case. However, the MHT's deferred decision making is complicated by the fact that the overall tracker is producing a blended estimate as well as individual Williams filter estimates. To preserve the deferred decision making capability in the blended estimate, it was found that modal probabilities should be lower-bounded in IMM as well as MMAE.

Consider an IMM with no lower bounding of the modal probabilities. As shown in Eq. (2.51), the modal probability calculated at instant (k), conditioned on information through instant $(k-1)$, is a function of the modal probability from the previous time instant $(k-1)$. Normally, the mixing of estimates at the end of each cycle is designed both to increase agility and to eliminate the need for *ad hoc* modifications like lower-bounding. In cases without measurement association uncertainty,

a particularly large measurement residual in a given filter at a given sample instant is a relatively good indication that the filter is not the best match to truth. However, in cases with measurement association uncertainty, the conditional pdf's in the denominators of Eqs. (2.51) and (2.53) are computed using a pseudo-residual. Because the pseudo-residual is from the various measurement associations, many of which may be clutter-originated measurements, the *instantaneous* pseudo-residual must be considered with care. A large instantaneous pseudo-residual could drop a filter's modal probability extremely low in a single sample period, causing this filter to be under-emphasized in the *mixing* of the estimates.

Remember that the deferred decision making within the MHT is taking place in the hypothesis (or, mixture component) reduction algorithm, which is within *each* elemental Williams filter in the multiple-model case. During the IMM's mixing process, pseudo-states and pseudo-error covariances are mixed as a probability-weighted sum and passed to each filter as an initial condition for the next sample period. Those mixing probabilities are a function of the filter's modal probability (see Eq. (2.47)). If an instantaneously large pseudo-residual forces a modal probability to be extremely low, that filter will receive a mixed initial condition at the next sample period weighted heavily towards the *other* filters in the system. In effect, the IMM mixing short-circuits the MHT's deferred decision making by declaring all of the components within the low probability filter to be of little benefit at the next sample period. By lower-bounding the modal probabilities in an IMM, this effect is somewhat mitigated because it prevents any single filter from being totally overwhelmed by the other filters in the mixing process.

4.2.2 Measurement Gate Unions. In Chapter 2, Equation (2.22) shows the joint density of the state $\boldsymbol{x}(k)$ and the uncertain model parameter $\boldsymbol{a}(k)$, conditioned upon the measurement history $\boldsymbol{Z}^k$. In a Kalman-filter-based MMAE or IMM, on a given sample period, each filter receives the same measurement and produces a measurement residual using that measurement and that filter's current state estimate.

In the MHT algorithms using the Williams filter, all measurements must be gated before they are used to form new association hypotheses. Only after gating, therefore, are measurements actually used in a measurement update of each component in the mixture. However, as discussed in Williams [30], the gate is formed around each component in the mixture. Implicit in that definition is the fact that each component, in each filter, will have a different gate and may allow a different measurement or group of measurements to pass inside the gate.

This poses a significant problem, because the MMAE and IMM algorithms involve conditioning upon a measurement history through sample (k), but measurement gating creates a situation in which each component, and more importantly, each filter may compute measurement residuals,

modal probabilities, and component updates using *different* measurement histories. Even though the developments of Kalman filters and multiple-model algorithms are conditioned upon a *particular* measurement history, they are conditioned on a single, unique measurement history. The problem caused by measurement gating in the MHT is obvious if one considers what would happen if each filter in a Kalman-filter-based MMAE or IMM received a different measurement on the same sample period. If each of these different measurements were relatively close to the filter-propagated state estimate, the measurement residual would be small in all filters, and the modal probabilities would be essentially meaningless.

The obvious solution is to ensure all filters in the MHT receive the same measurements, and one means of accomplishing this is a union of all measurement gates. In theory, one can envision a union of gate "sets" in which the overall gate for the system is the union of the sets defined by each component's individual measurement gate. In practice, this would be extraordinarily difficult to implement, so instead, the algorithms used in this research undergo a two-pass gating process. On the first pass, each component in each Williams filter gates measurements according to its own gate as determined in the algorithm by Williams [30]. The details of this gate computation are not vitally important to the discussion at hand, but it should be mentioned that each component's gate is a rectangular box formed around its predicted measurement with box side length calculated as a function of the maximum eigenvalue of the measurement residual covariance matrix.

On the second gate pass, each measurement (in the overall measurement set) that passed a component's gate is collected into a unioned set. Using a simple indexing technique, each component in each Williams filter is then associated with each measurement *included in the unioned set but not gated by that component on the first pass.* This ensures that every component in every Williams filter forms an association (and measurement residual) with every measurement included in the overall system, and restores the proper and necessary conditioning of the state and filter probabilities upon a single, and unique measurement history.

The importance of gate unioning cannot be understated. It does, however, dramatically increase the number of measurement associations processed at each cycle for a given clutter density. For example, assume a system has N_f elemental filters, and the m-th filter gates $N_z(m)$ measurements on a particular cycle. Also assume each filter had $N_r(k-1)$ pre-existing components from which to form measurement associations and none of the first-pass gates overlap. The resulting number of associations within the m-th filter will be:

$$N_r(k-1) \cdot \left[\sum_{i=1}^{N_f} N_z(i) \right] + N_r(k-1) \tag{4.1}$$

where the second $N_r(k-1)$ term is the number of missed-detection components and are not associated with any measurement. If

$N_f = 3$
$N_r(k-1) = 4$
$N_z(m) = 5$ for $m = 1, \ldots, 3$

then each filter will have 64 association history hypotheses prior to hypothesis reduction. In the three-filter system, this means there will be 192 hypotheses existing within the entire MHT prior to reduction within each filter. If one assumes that the 15 unique, gated measurements in this example represent *all* measurements in the measurement space, then it becomes clear that even low clutter densities create a significant computational burden for the algorithm. Furthermore, each association history hypothesis as implemented in the software written for this research maintains several double-precision data matrices in addition to the minimal required state, error covariance, and probability weight. This results in a significant RAM requirement in addition to the obvious computational burden.

Despite the negative effects gate unioning has on algorithm speed and memory efficiency, the need to ensure a consistent measurement set across all filters made gate unioning a necessity. At low clutter levels (fewer than 15 measurements generated per cycle in the entire measurement space), the algorithm performed quite well, albeit too slow for a realtime application with one measurement update per second.

Finally, a modification was made to the gate sizing algorithm developed by Williams for use as the first-pass gate computation in this research. If, on a particular measurement cycle, an elemental filter finds *no* measurements in *any* of its components' gates, one could call into suspicion the validity of the modal probability computations. In such a case, the filter is only using measurements from the unioned set, which are in general, rather far from the filter's pseudo-estimate position (see SubSection 4.2.3 for a related discussion). To remedy this problem, a dynamic gate size increase occurs if no measurements are found on the first gate application, and successive increases take place until at least one measurement appears in the gate of at least one component in a given filter. Up to ten successive increases can take place before the filter uses only the unioned measurements.

Chapter 5 will outline, in further detail, some proposed alternatives to gate unions (such as a convex hull of individual gates, or a simple approximation thereto) and successive gate size increases.

4.2.3 Lower-Bounding of Component Probabilities. One of the side effects of gate unioning is that measurement associations formed using measurements from the unioned set (i.e., those measurements not within the filter's own first-pass gated set) will often have very large measurement residuals. Furthermore, these measurement residuals may be far outside the one-sigma bound of the measurement residual covariance defined by the component with which it is matched. Since

the new association history hypothesis' probability is computed using the $[\boldsymbol{r}^{\mathrm{T}}\,\mathbf{A}^{-1}\boldsymbol{r}]$ term for the pre-existing component and its associated measurement, measurements from the union set will often create association history hypotheses with very low probabilities. Unfortunately, it is the entire measurement set that gives meaning to the multiple-model algorithm, and low probability components will be prime targets for pruning or merging during hypothesis reduction. The pruning or merging of hypotheses is desirable, for the most part, but not if it removes all of the components that would lead the multiple-model algorithm to believe a particular filter is too far from truth. If these low probability components are removed prematurely, the effective set of measurements used in each filter is only that from the first-pass set, and the algorithm may assign diluted probability weights to all filters.

Lower-bounding of the component probabilities ensures that the hypothesis reduction algorithm does not dispose of components that could be useful in the probability computation process. Careful choice of a lower-bound value is critical, because the lower bounding is applied prior to hypothesis reduction when the number of new components could be as high as several hundred. A bound which is too high will prevent the hypothesis reduction algorithm from merging and pruning effectively, but a bound too low will not create the desired effect. In this research, the lower bound was set at 5.0×10^{-5} for all clutter density cases.

4.2.4 Filter Restart and Mixing with Gaussian Mixtures. In a Kalman-filter-based MMAE, filter restarts are accomplished using the MMAE blended state estimate $\hat{\boldsymbol{x}}_{blend}(k|k)$ and error covariance $\mathbf{P}_{blend}(k|k)$. In an IMM, initial condition mixing is accomplished as a probability weighted sum of individual filter state estimates $\hat{\boldsymbol{x}}_m(k|k)$ and error covariances $\mathbf{P}_m(k|k)$ for $m = 1, \ldots, N_f$. In the actual reset or initial condition mixing, there is a one-for-one replacement of the old value with the incoming value. But, each filter within the MHT is carrying, at every sample time, a Gaussian mixture with an arbitrary number of components. In order to accomplish filter restarts or filter estimate mixing, there must be some means of rectifying the single blended estimate (MHT MMAE) or filter pseudo-states and pseudo-covariances (MHT IMM) with the Gaussian mixture.

The arrived-upon solution was dubbed "re-centering the cloud", where the cloud is the set of all component state estimates with a given filter. The center of the cloud is considered to be the filter's pseudo-state, $\hat{\boldsymbol{x}}_m(k|k)$ for the sample period of interest. Upon a reset or mixing, the pseudo-state is subtracted from the state estimate of each component, yielding a mean-spread term for each component. The mean-spread term for each component is then added to the incoming state estimate, forming a new "cloud" with "center" defined by the incoming state estimate. The error covariances for each component were set to the error covariance of the incoming estimate.

This solution not only worked well, but it entails minimal computational overhead. All mathematical operations are matrix additions and subtractions, and the covariance term involves only a simple replacement. In reality, if the algorithm were to recompute the pseudo-error covariance, it would arrive at a different value than the incoming "replacement" error-covariance due to mean-spreading. This recalculation was deemed too computationally expensive, and the replacement technique did not create any noticeable performance degradation.

4.3 Ultra-Low Clutter Results

As a verification of the MHT algorithms, a few filter suites and truth scenarios were run under "ultra-low clutter density". This equated to an expected 2 measurements/cycle, and since the probability of detection was set to 1.0, at least one measurement was always the true measurement. The maximum number of components existing after the Williams mixture reduction cycle was specified to be 10. It was anticipated that, at such a low clutter density, the MHT performance under these circumstances would come rather close to that of the Kalman-filter-based MMAE or IMM for the no-clutter case. Theoretically, the performance of the MHT algorithm should improve as clutter density decreases, although the performance improvement was less than expected.

There are two primary limitations that prevent the MHT algorithms under very low clutter from performing as well as the Kalman-filter-based algorithms working with no measurement association uncertainty. First, the Williams filter will begin merging and pruning operations as soon as the number of components entering the reduction cycle exceeds the defined upper bound on the number of components leaving the reduction cycle. If the system indeed generated exactly two measurements at every cycle, one would expect merging and pruning to take place at $2^k = \text{MaxNumComp}$, where MaxNumComp is the maximum number of components leaving the cycle, and (k) is the sample period. At $k = 6$, each filter will have $2^6 = 64$ components prior to reduction, and if MaxNumComp is set to 50, then merging and pruning would first take place at this sample period. For these tests using ultra-low clutter density, the MaxNumComp parameter was set to 10.

Prior to merging and pruning, there is guaranteed to be at least one association history hypothesis that includes *only true measurements*. This association history hypothesis could exist in more than one filter, because each filter would have a component that is continually matched with the real measurement even though its propagation dynamics differ in each filter. Once merging and pruning take place, there is no guarantee that the hypothesis consisting only of true measurements will continue to exist uniquely. It is possible that other hypotheses could be merged with it, at which point the new component no longer matches anything within a Kalman-filter-based algorithm at the same cycle.

The second limitation is caused by the gating computations. As discussed at length in Section 4.2.2, the true measurement could fall outside the gate of any or all of the components with an MHT elemental filter. At that point, that filter is at a distinct disadvantage, because the true measurement may only exist in the unioned set. In a worst case situation, the measurement may not be present in *any* gates in the system. The major issue to consider is that the Kalman-filter-based algorithms never exclude a measurement from an update, whereas the MHT algorithms may exclude one or more measurements in the gating process.

Even under the assumption that the target's true measurement was within at least one gate somewhere in the system, Section 4.2.2 explained how measurements from the unioned set may form hypotheses with unrealistically small probability weights. Once merging and pruning have begun, even a component consisting only of associations with the true measurement may be pruned because its probability is too low. In contrast, the Kalman-based algorithms never gate measurements, so each filter is always assured equal access to the true measurement. Chapter 5 will recommend further analysis of gate sizing to reduce the impact this has on the MHT algorithms at all clutter densities.

The best configurations for this verification process are the single-filter, IMM with Identity Markov cases. These behave exactly as a dedicated single-filter algorithm, and the problems associated with some filters having no measurements in their gates is mitigated. Suite 1, a single-filter CV model, was run against Scenario 1, the time-invariant CV truth model. Figure 4.46 shows the single-filter output from this tracker, and it appears the be performing quite well (Fig. 4.1 shows the same runs for the no-clutter case). Its actual state estimate error is within the filter-computed standard deviations (difficult to see on the printed versions of these plots) and the $\pm 1\sigma$ error lines are on the order of 2 m in position and 2 m/sec in velocity.

The FOGMA filter did not do as well, and on at least one run, the single-filter tracker was divergent. Figures 4.47 and 4.48 show the single-filter FOGMA tracker has at least one divergent state estimate. Monte Carlo run number 4 is one of the divergent cases. The top pane of the single-run plots show the blended (or elemental filter, as the case may be) state estimate y-vs.-x position in meters, with the blended (or filter) estimate denoted by black circle data markers and the truth model trajectory denoted by grey star data markers. The large triangle indicates the initial truth trajectory position and the large square indicates the final truth trajectory position. The lower panes show x-position and y-position error in meters versus time.

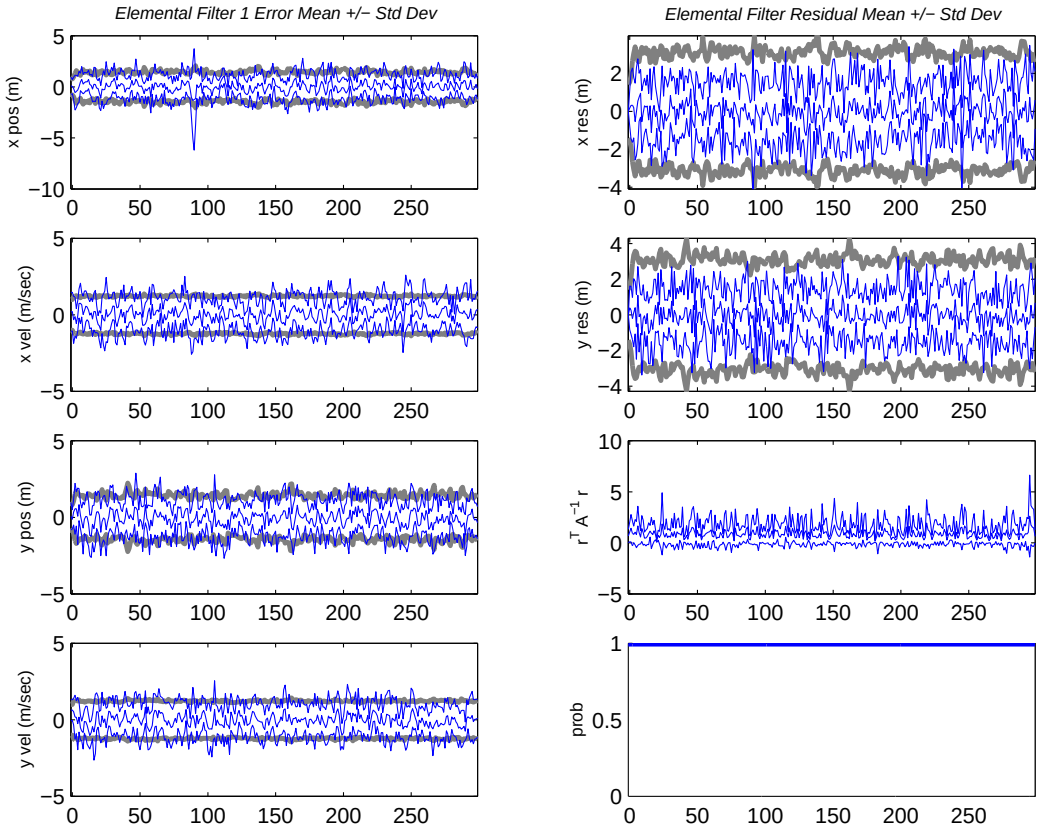

Figure 4.46: A single, CV filter MHT running against a matching truth model in the ultra-low clutter density case. (Suite 1 vs. Scenario 1: IMM (Identity Markov))

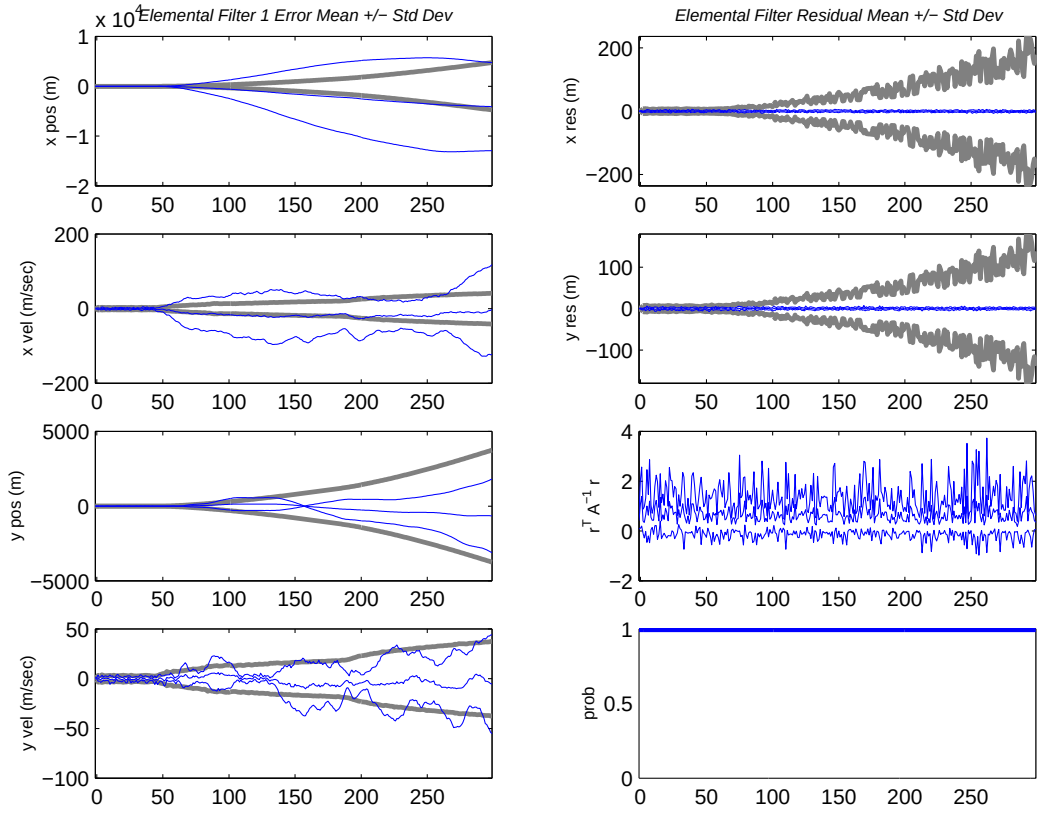

Figure 4.47: A single, FOGMA filter MHT running against a matching truth model in the ultra-low clutter density case. (Suite 2 vs. Scenario 2: IMM (Identity Markov))

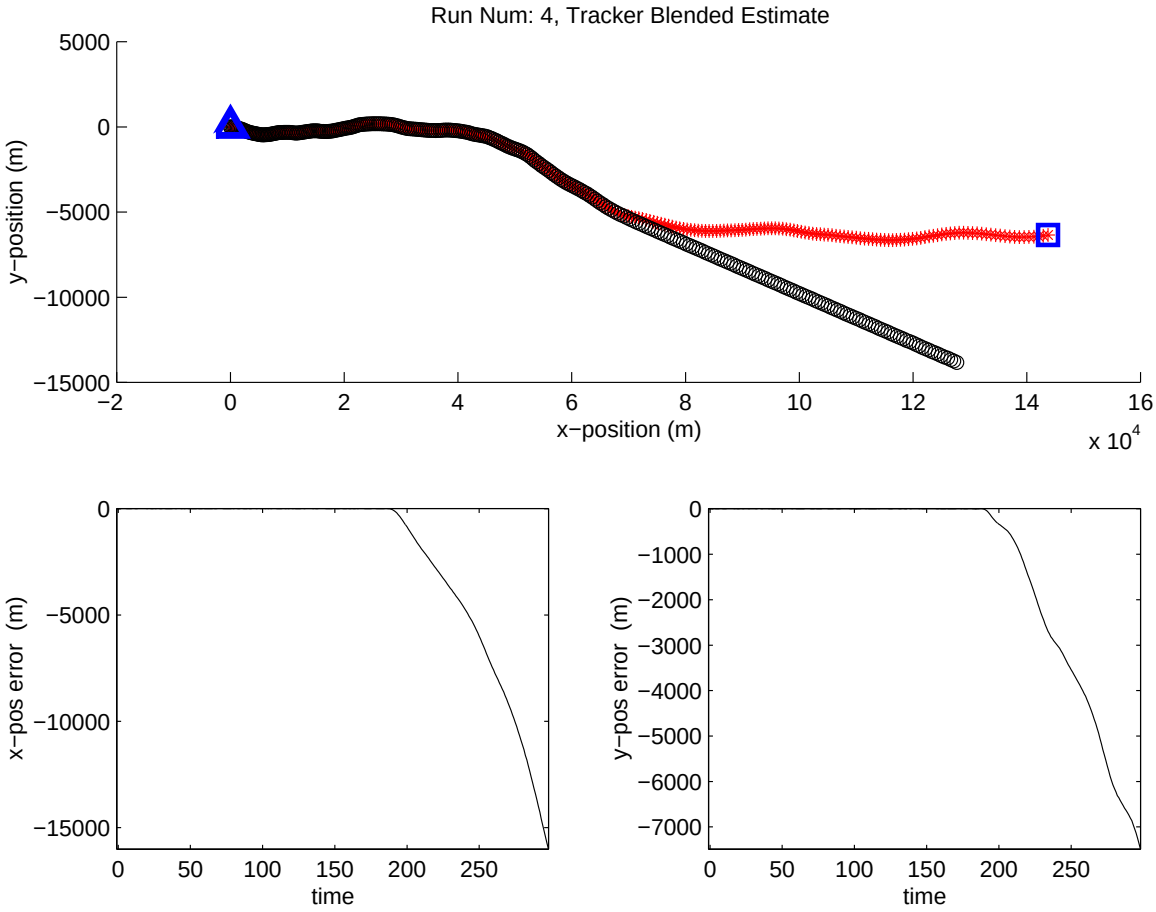

Figure 4.48: Blended estimate on run number 4, from a single FOGMA filter MHT in the ultra-low clutter density case. (Suite 2 vs. Scenario 2: IMM (Identity Markov))

Suite 4, a two-filter configuration of all FOGMA models, was tested against the maneuvering truth model in Scenario 4. Also, Suite 5, containing two TPV filters and one CV filter, was tested against the TPV maneuver Scenario 5. In both of these trials, the tracker estimates looked more like the estimates seen in the higher clutter cases than they did the estimates seen in the Kalman filter cases.

In summary, the ultra-low clutter density case provided somewhat ambiguous results except for the case of the single CV filter against a matching truth. If the software allowed more flexibility in the gate size determination, it is likely that other configurations would begin to perform more closely to the Kalman filter configurations.

4.4 Low-Clutter Results

All filter suites and truth scenarios were tested using a clutter density that produced an expected five measurements per cycle with a probability of detection equal to 1.0. The maximum number of components existing after the Williams mixture reduction cycle was specified to be 30. This section will evaluate the results of those tests in essentially the same progression as that used for the Kalman-filter-based results. Slightly more emphasis will be given to the single-run tracking solutions, because they provide insight into tracker behavior that cannot be ascertained from the Monte Carlo statistical plots. Furthermore, the MHT algorithms experienced track divergence more often than the Kalman-based algorithms, so the statistical plots must be interpreted more carefully. Lastly, the reader is reminded that the track loss checks discussed in Chapter 3, Section 3.4 will be treated in Appendix C. The emphasis here will be on qualitative analysis of tracker performance, whereas the appendix emphasizes a more quantitative approach to the analysis.

4.4.1 Single-Filter Model Against Benign Truth. These cases should provide the best track performance, because the truth models and filter models match for all time. Since no aggressive changes in truth trajectory take place, these filters should have the least amount of trouble maintaining a valid target track for the course of the simulation. As was the case in the Kalman-filter-based algorithms, trials involving only a single elemental filter were only conducted using an IMM with Identity Markov matrix, because that configuration replicates the performance of a dedicated, single-filter algorithm.

The plot of the single Monte Carlo run (Fig. 4.49) shows that the MHT filters are capable of maintaining a track in the presence of measurement association uncertainty, although the ability to maintain a track generally depends on the *particular* measurement set received on a given Monte Carlo run. That is to say, some trials may create clutter measurement patterns that cause the filter

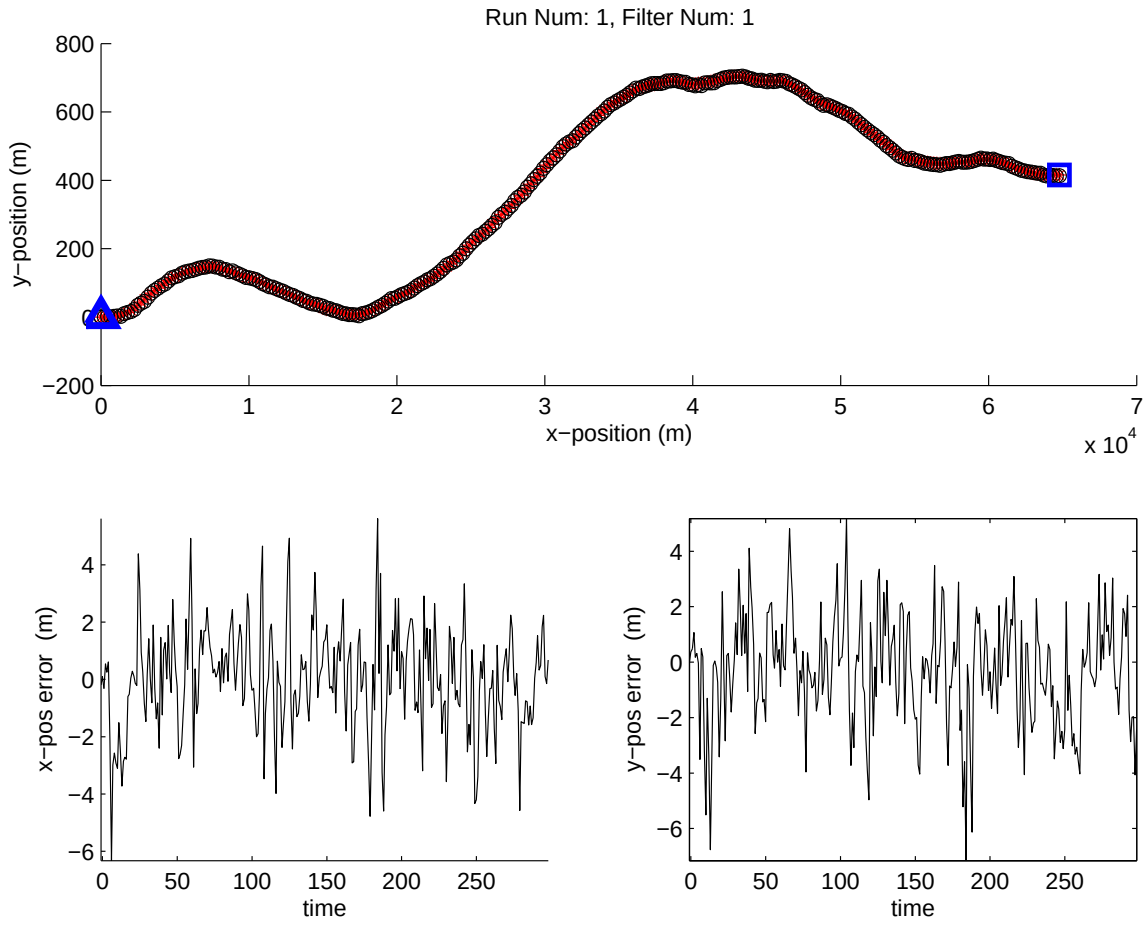

Figure 4.49: A single Monte Carlo run of a single CV filter MHT against a CV truth with no maneuvers. (Suite 1 vs. Scenario 1: IMM (Identity Markov))

---

to diverge and never recover, while other trials may create clutter measurement patterns that do not impede the tracker's ability to determine the true target trajectory.

Figure 4.50 indicate the single constant-velocity filter performed very well against a constant-velocity truth. Like the Kalman-filter-based algorithm output in Fig. 4.1, the filter state estimate error is relatively small. The MHT appears, however, to compute pseudo-residual covariances that are substantially worse than the actual residuals, as indicated by the wide gray lines in the x and y residual plots (first and second plots, right column).

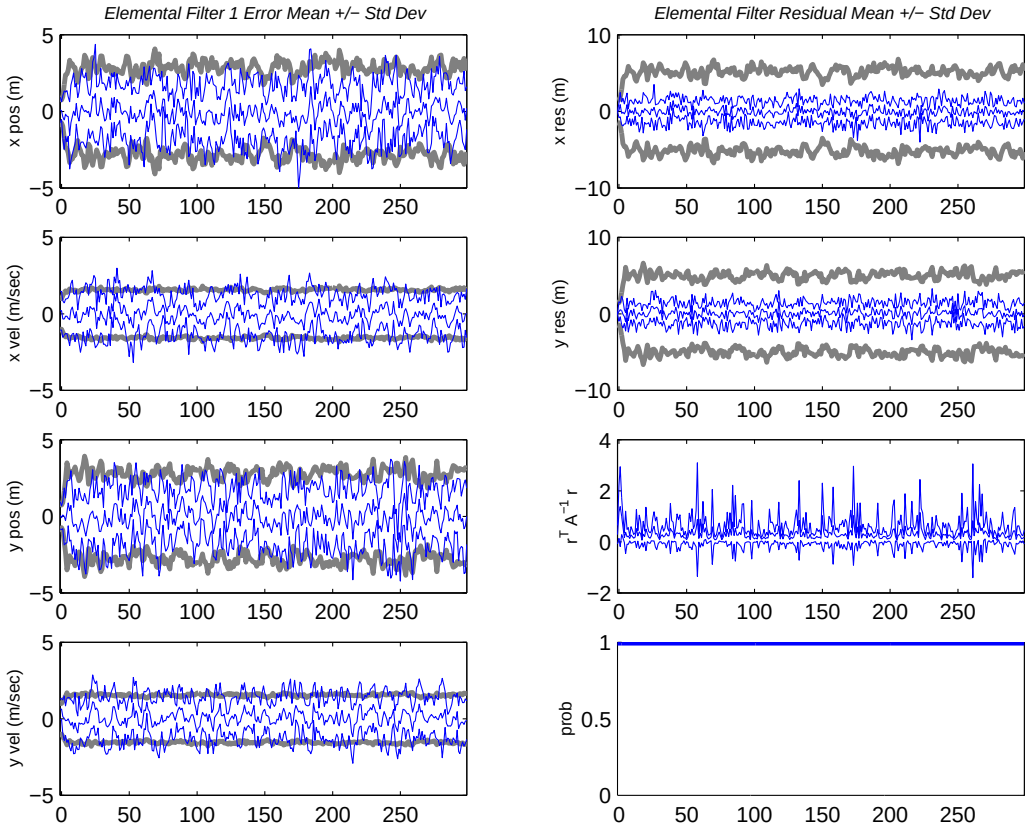

Figure 4.50: Output from the single CV filter MHT containing data from 10 Monte Carlo runs. (Suite 1 vs. Scenario 1: IMM (Identity Markov))

Figures 4.51 through 4.53 show the results for a single, benign FOGMA filter running against the benign FOGMA truth scenario. The single Monte Carlo run plot in Fig. 4.51 shows the tracker is maintaining target track on the first Monte Carlo run. Figure 4.52 shows that the filter was divergent on run number 4, and the error committed on this run contributes significantly to the divergent behavior shown in Fig. 4.53. It should be noted that this filter was divergent (i.e., lost track and never recovered) on runs 4, 6, and 8 out of 10 Monte Carlo runs. Overall, the performance of the tracker is very good when the filter is maintaining track, building confidence in our assumptions about the Bayesian MHT problem as well as building confidence in our algorithm. On the other hand, since this is a single filter suite running against a matching truth scenario, the expectation is that the tracker should maintain target "lock" on all Monte Carlo runs just as the constant-velocity, Suite 1 versus Scenario 1 case does. More worrisome is that fact that Fig. 4.53 shows very small pseudo-residuals for all time, even though a Kalman-filter-based tracker with the same tracking behavior would exhibit very large residuals during periods in which many filters are divergent. In this case, the last 100 seconds or so of simulation should exhibit increasingly large $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ mean $\pm 1\sigma$ values if the algorithm were operating exactly as a Kalman-based MMAE or IMM. To the contrary, Fig. 4.53 shows the pseudo-residuals getting smaller as time progresses, and this behavior cannot be explained, beyond pointing out that the filter-computed pseudo-residual covariances (first and second panes, rightmost column) are getting very large as the simulation progresses. This would certainly cause the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ term to become very small even if the pseudo-residual itself is not changing or growing at a slower rate.

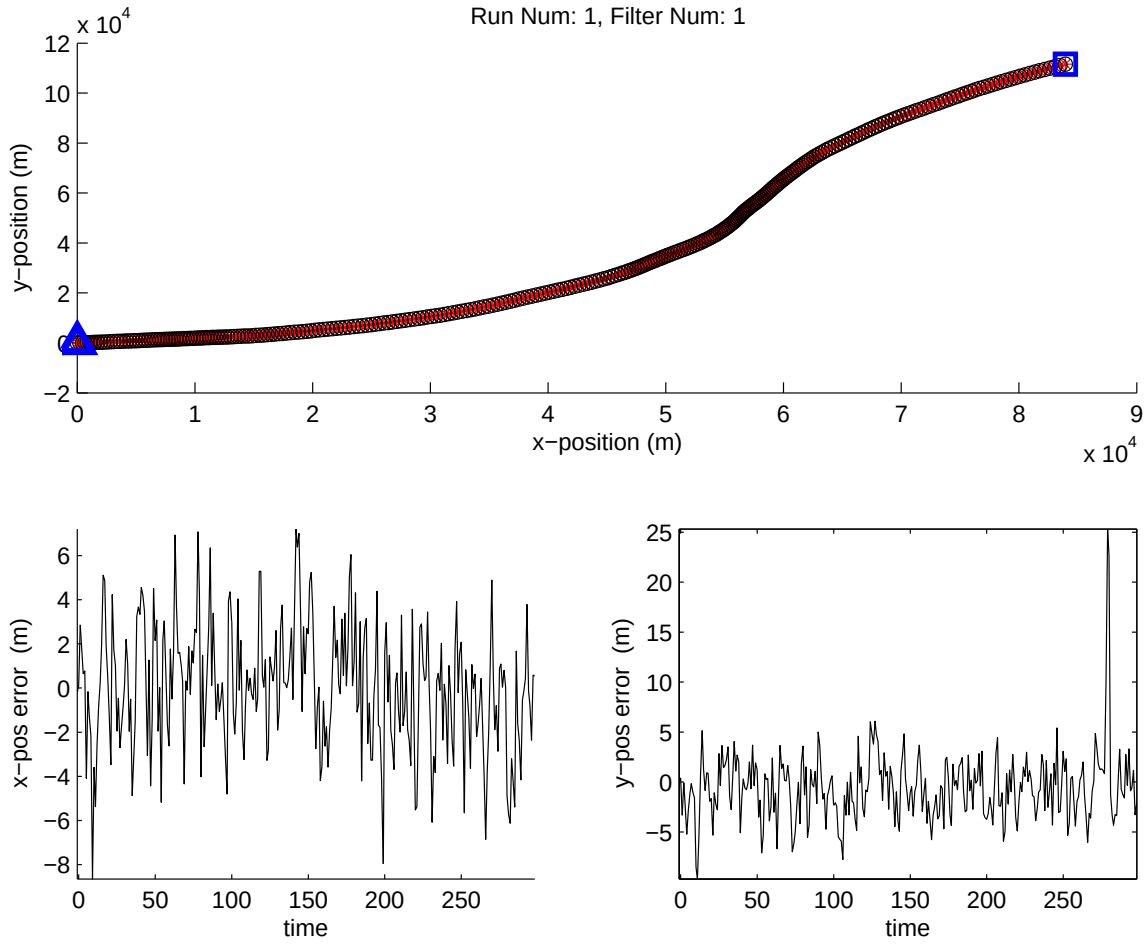

Figure 4.51: A single Monte Carlo run (run number 1) of a single FOGMA filter MHT against a FOGMA truth with no maneuvers. (Suite 2 vs. Scenario 2: IMM (Identity Markov))

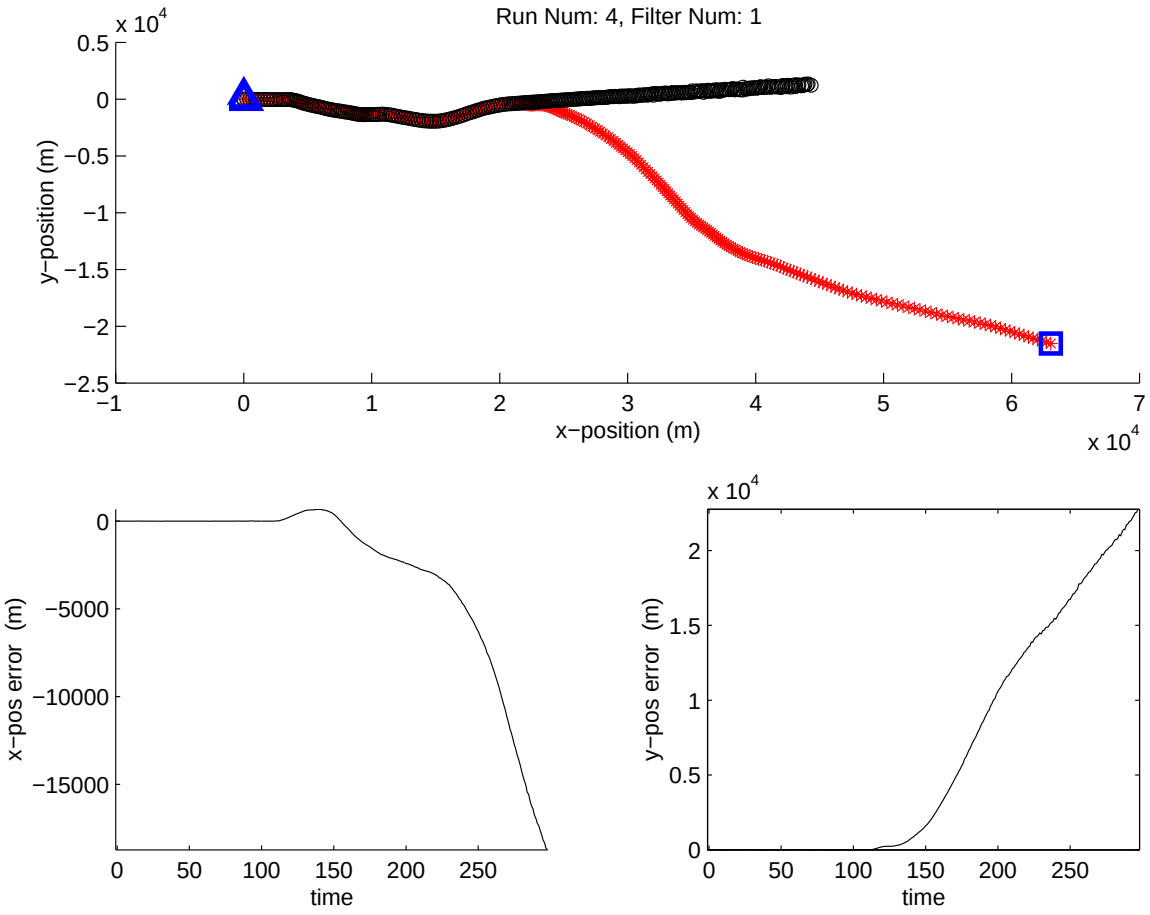

Figure 4.52: A single Monte Carlo run (run number 4) of a single FOGMA filter MHT against a FOGMA truth with no maneuvers. Note the filter is divergent on this run. (Suite 2 vs. Scenario 2: IMM (Identity Markov))

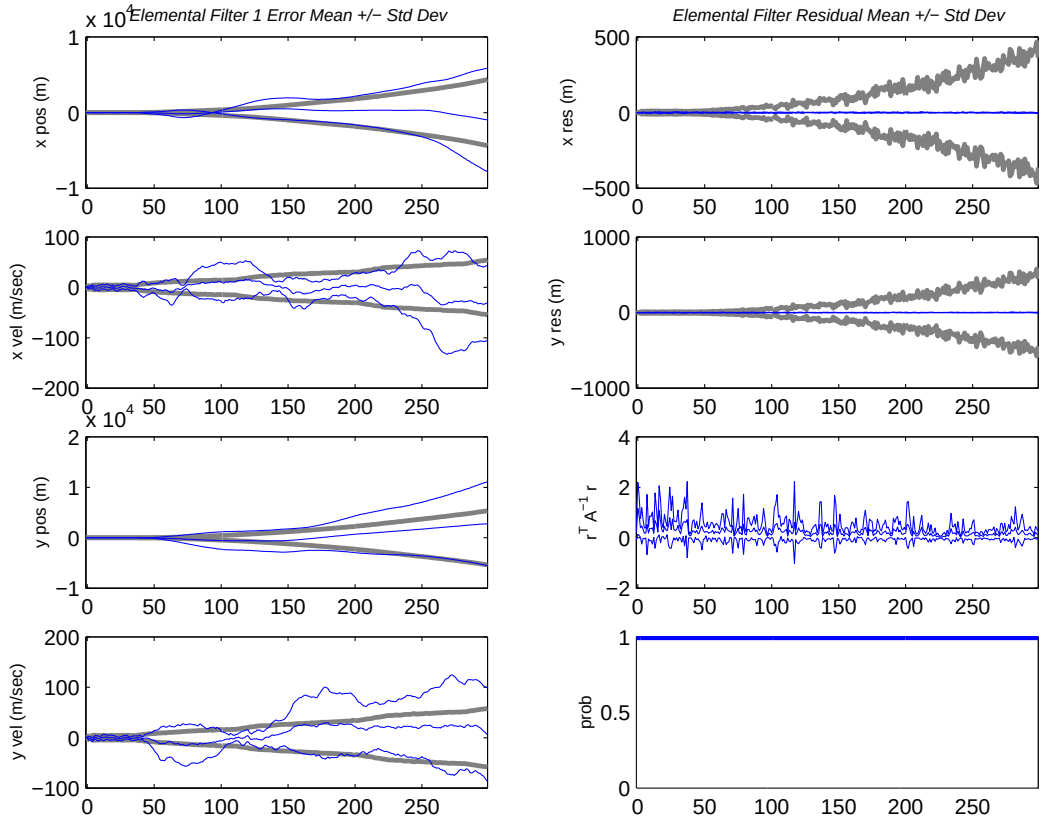

Figure 4.53: Output from the single FOGMA filter MHT containing data from 10 Monte Carlo runs. (Suite 2 vs. Scenario 2: IMM (Identity Markov))

4.4.2 Single Filter Against Time-Varying Truth Model. The Kalman-based algorithms showed that single filters will have more difficulty tracking a time-varying truth model than they do a matched, and time-invariant truth model. Given the fact that the single FOGMA filter had some difficulty tracking a matching dynamic, time-invariant FOGMA truth model, it is clear that MHT trackers should exhibit less predictable performance than the equivalent Kalman-based tracker.

Figures 4.54 and 4.55 show results from a single, benign FOGMA filter against the time-varying FOGMA truth model of Scenario 4. Figure 4.55 may be compared against Fig. 4.3, which is the no-clutter case. Again note that divergent behavior is shown in the summary of the 10 Monte Carlo runs, and the single run plot provides an example of the filter output. In this case, track was maintained only on run number 10 – all other runs experienced track loss shortly after the onset of the maneuver at ($k = 100$).

4.4.3 Configurations with Two FOGMA Filters. The MMAE using filter Suite 4, containing both a benign and aggressive FOGMA filter, running against truth Scenario 4 is a case in which the MHT tracker perform exactly as desired, although the track error performance was slightly worse than anticipated. The results for the MMAE appear in Figs. 4.56 through 4.60, and these can be compared with the no-clutter cases in Figs. 4.5, 4.6, and 4.7. The elemental filter probabilities in Fig. 4.59 are the same as those which appear in the individual elemental filter plots, simply collected together and magnified for easier viewing (this combined probability plot was not created for the no-clutter case).

Notice the probability weights do not adjust as quickly or assertively towards the correct elemental filter as they do in the no-clutter case. Again, the filter-computed pseudo-residual covariance appears to be overly pessimistic compared to the actual pseudo-residuals. The tracker estimate does recover after the cessation of maneuvers, indicating the deferred decision-making ability of the MHT is a major factor in performance. The jumps in the state estimate error mean $\pm 1\sigma$ plot of Fig. 4.58 are largely a result of the tracker "snapping" itself towards a newly recognized trajectory.

This snapping effect can also be seen in Fig. 4.60, appearing as discontinuities at approximately ($k = 160$), ($k = 190$) and ($k = 210$). Notice the error plots will show large spikes at times of filter estimate discontinuity, and remember that the leftmost pane is x-position error and rightmost is y-position error. The discontinuity at ($k = 160$) is not easily seen in the uppermost pane due to plot scaling, but the other two discontinuities are evident. These discontinuities are a result of the deferred decision-making within the elemental filter itself, which is precisely the behavior expected from the Williams-filter-based multiple-model architecture.

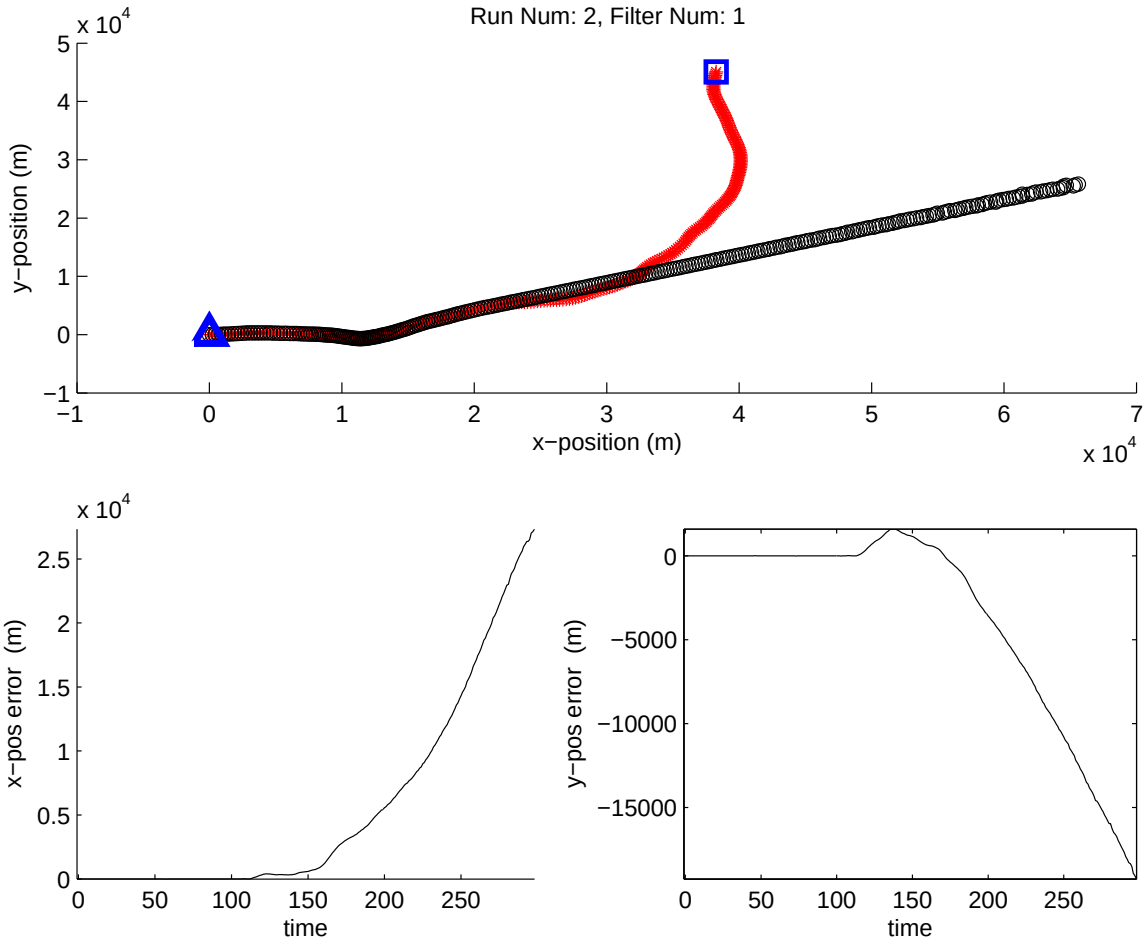

Figure 4.54: A single Monte Carlo run of a single FOGMA filter MHT against a time-varying FOGMA truth. (Suite 2 vs. Scenario 4: IMM (Identity Markov))

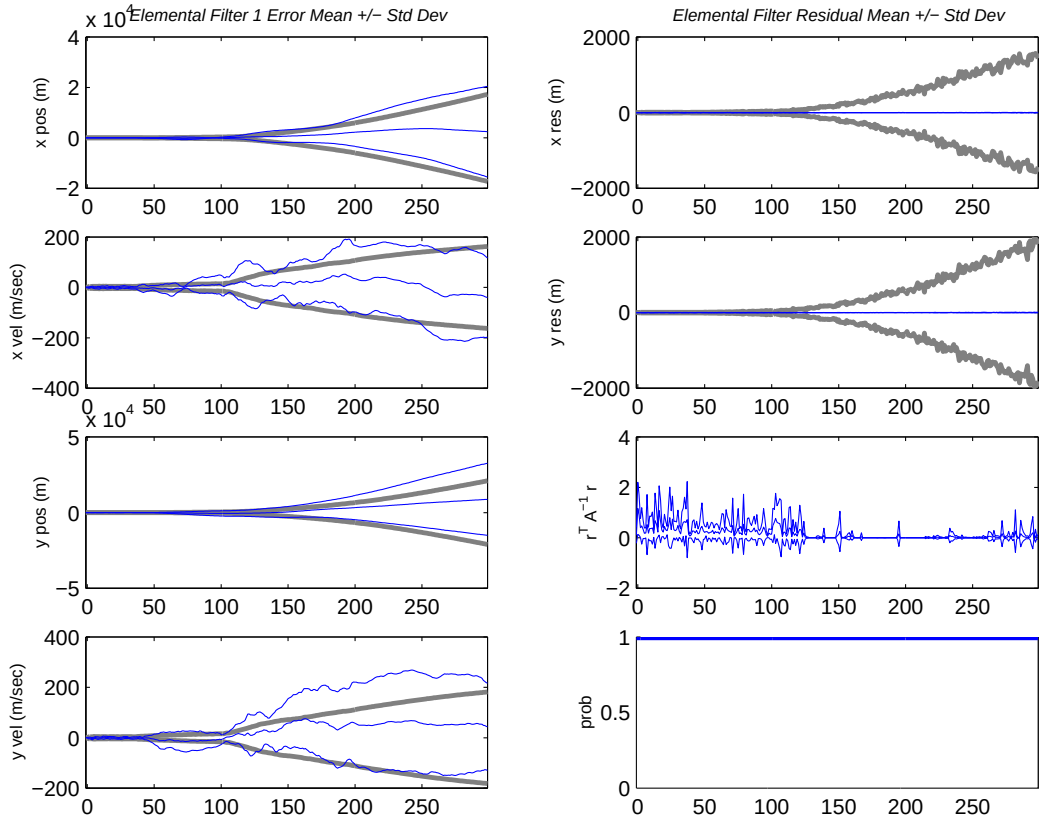

Figure 4.55: Output from the single FOGMA filter MHT containing data from 10 Monte Carlo runs where the truth model is a time-varying FOGMA. (Suite 2 vs. Scenario 4: IMM (Identity Markov))

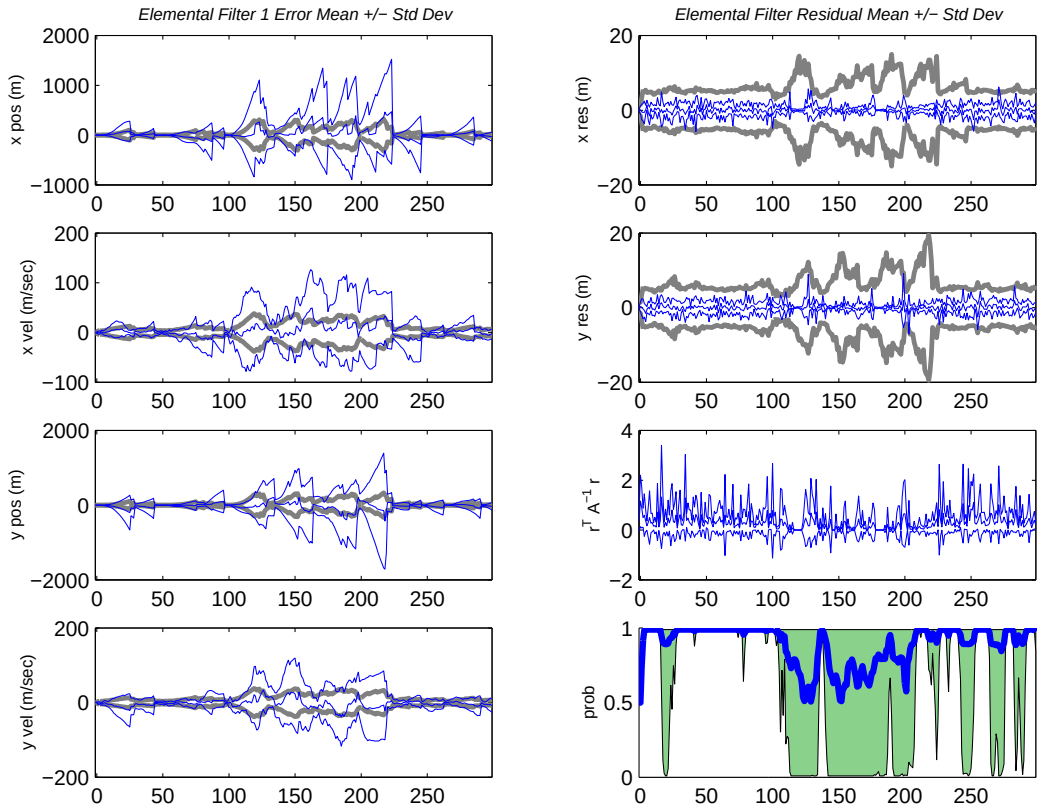

Figure 4.56: Filter 1 output. (Suite 4 vs. Scenario 4: MMAE)

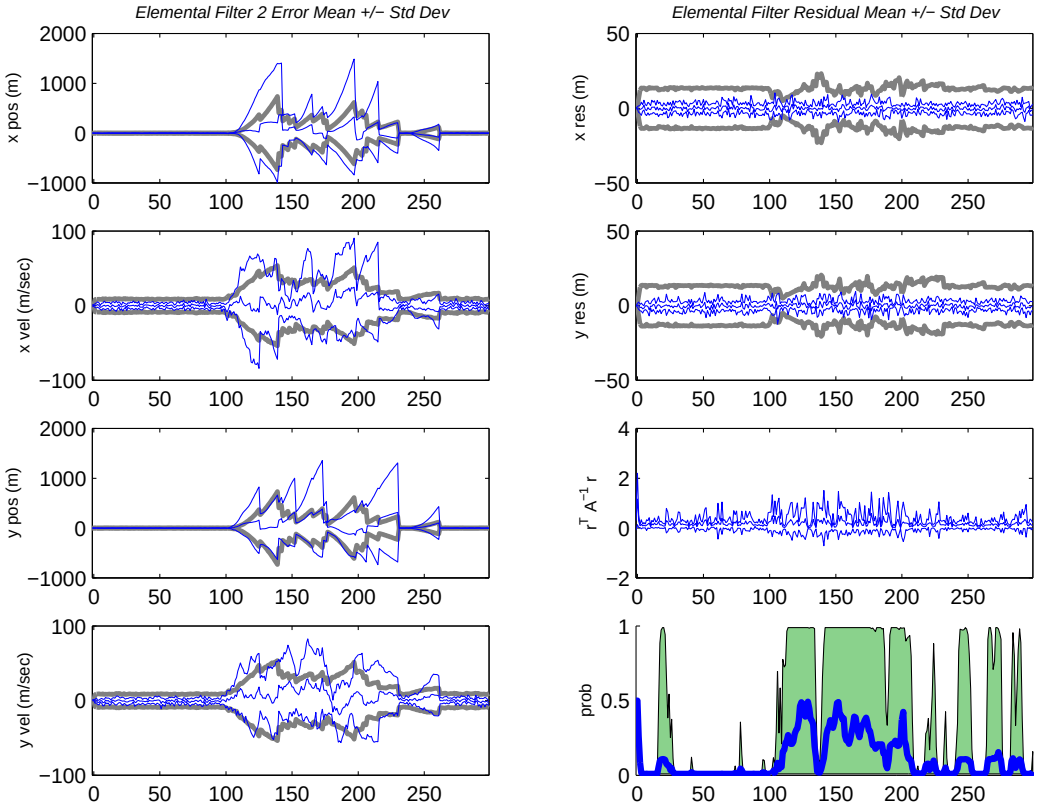

Figure 4.57: Filter 2 output. (Suite 4 vs. Scenario 4: MMAE)

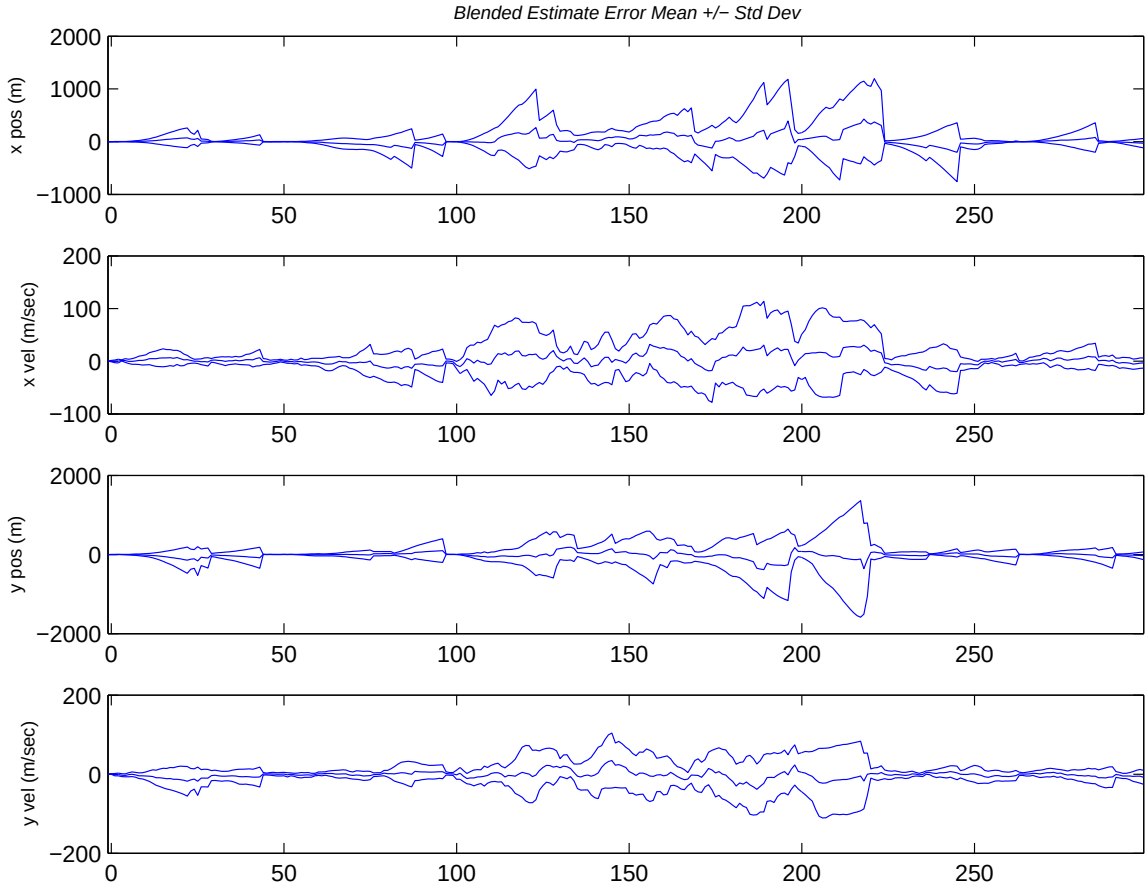

Figure 4.58: MMAE blended output. (Suite 4 vs. Scenario 4: MMAE)

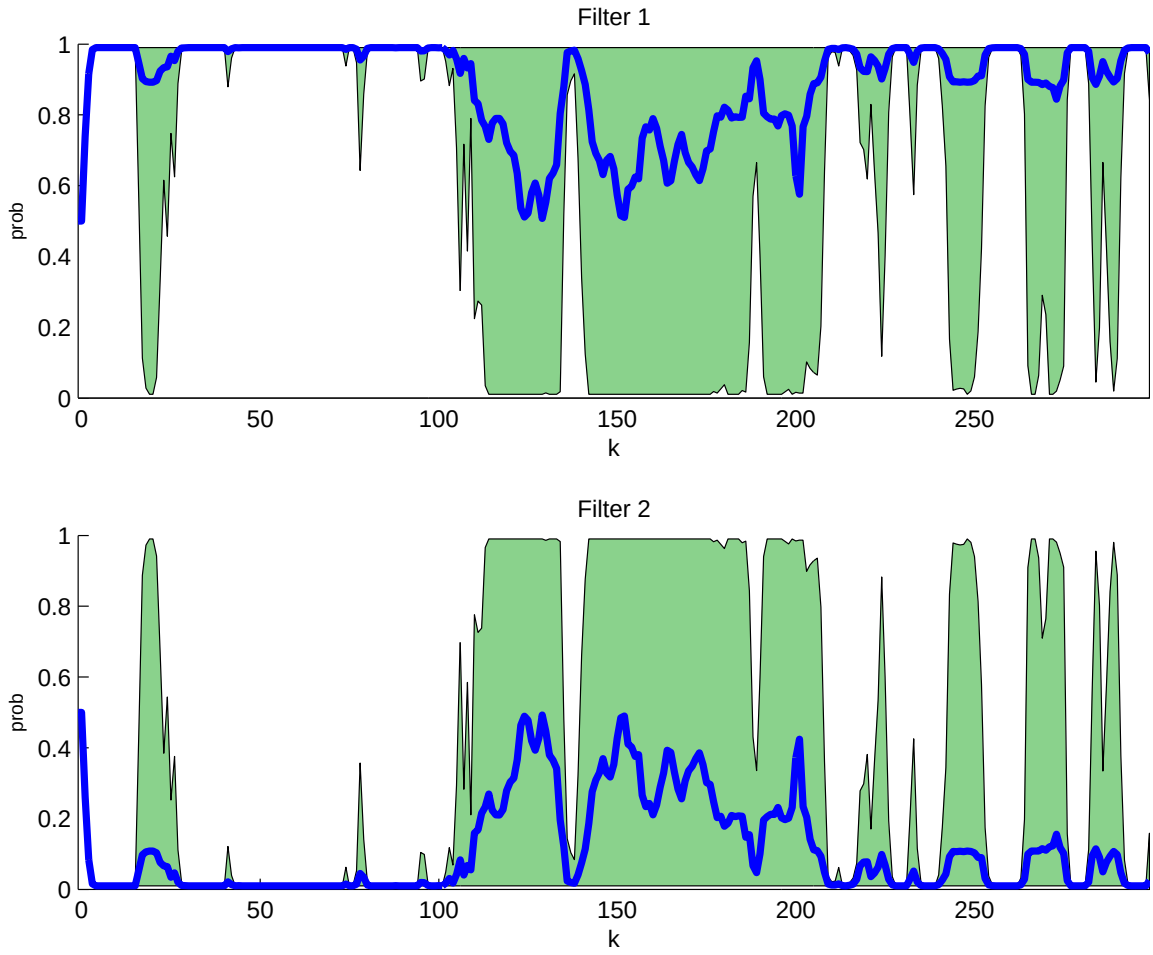

Figure 4.59: MMAE probability flow. (Suite 4 vs. Scenario 4: MMAE)

Similar discontinuities also appear in the estimate of filter 2 (Fig. 4.61), although the times of occurrence and magnitude differ. The error spike near ($k = 140$) in filter 2's estimate is worse than filter 1's error at the same sample time, but filter 2 matches the truth model dynamics during this phase of flight. Despite the matching of filter and truth model dynamics, it is clear that the MHT elemental filter estimate may *still* exhibit significant error for short periods of time as its deferred decision-making converges onto the most appropriate trajectory.

Notice the blended estimate in Fig. 4.62 is heavily driven by filter 1's estimate, as evidenced by the similarities between the blended and filter position error plots (bottom two panes of Figs 4.60 and 4.62). This implies that filter 1's probability was very close to 1.0 for most of this run. The probability flow plots in Fig. 4.59 *do* imply that, on many of the trials, probability weights shifted appropriately as truth model changes took place. On run number 1, it appears that elemental filter probability did not flow quite like expected, because the blended error should have resembled filter 2's error between ($k = 100, \ldots, 199$) as the aggressive FOGMA maneuver was taking place. Notice that filter 2's error during this phase is substantially better than filter 1's except for the spike near ($k = 140$).

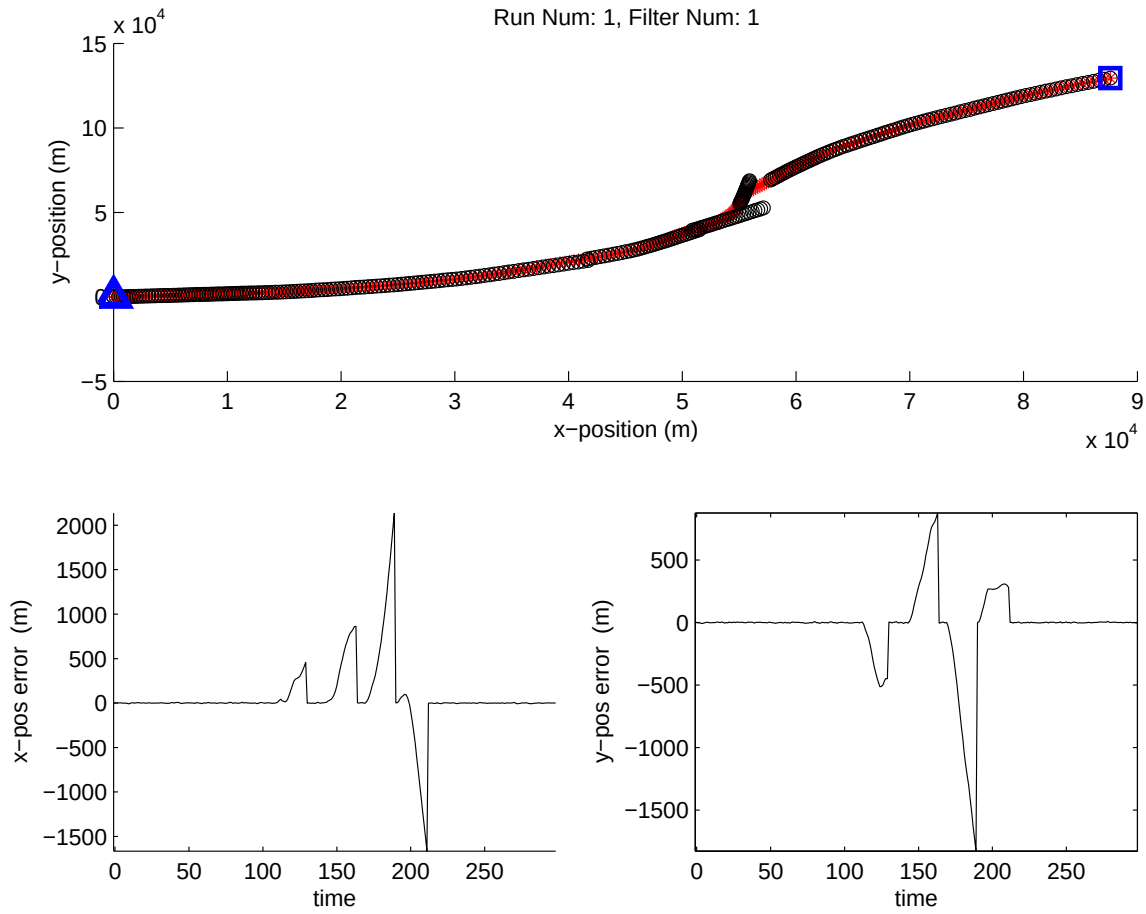

Figure 4.60: Filter 1 on run number 1. Notice the discontinuities in the track solution as deferred-decision making takes place. (Suite 4 vs. Scenario 4: MMAE)

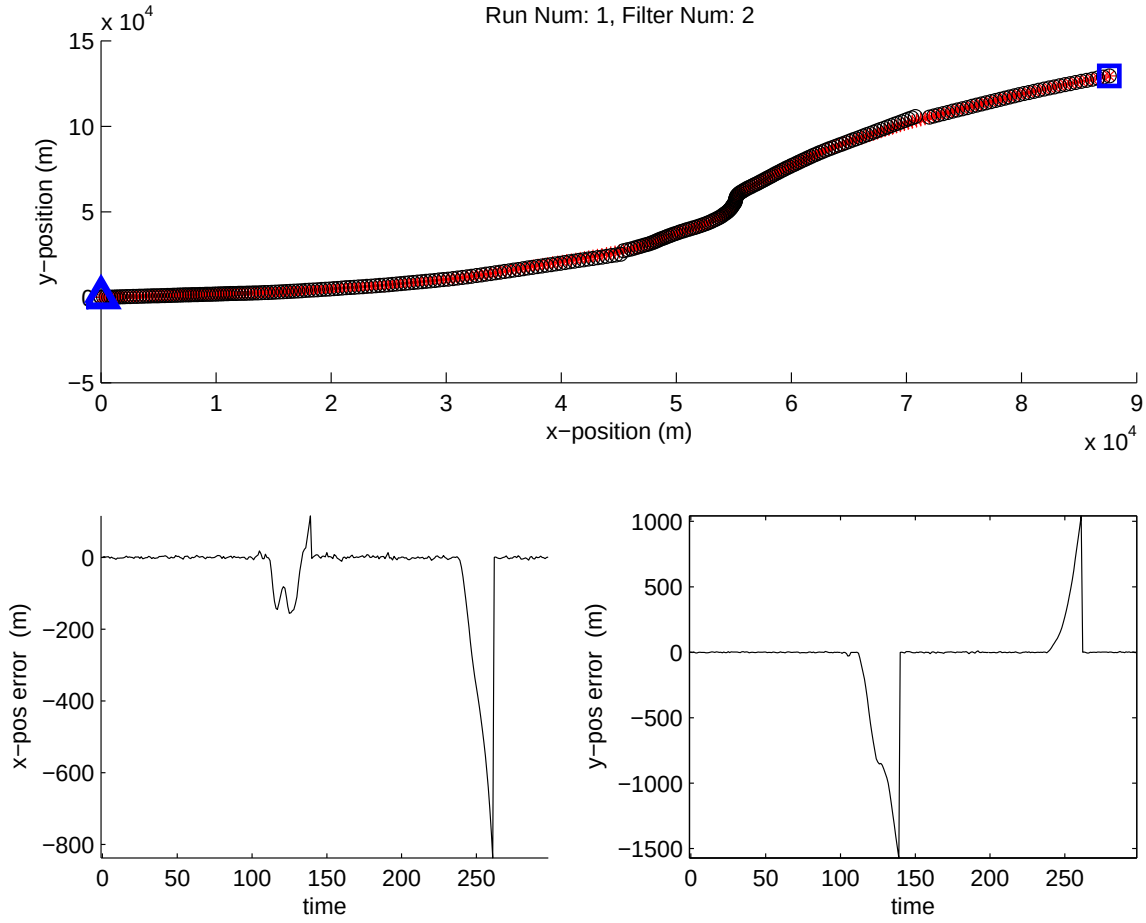

Figure 4.61: Filter 2 on run number 1. This filter matches the dynamics of the aggressive FOGMA maneuver from ($k = 100, \ldots, 199$). (Suite 4 vs. Scenario 4: MMAE)

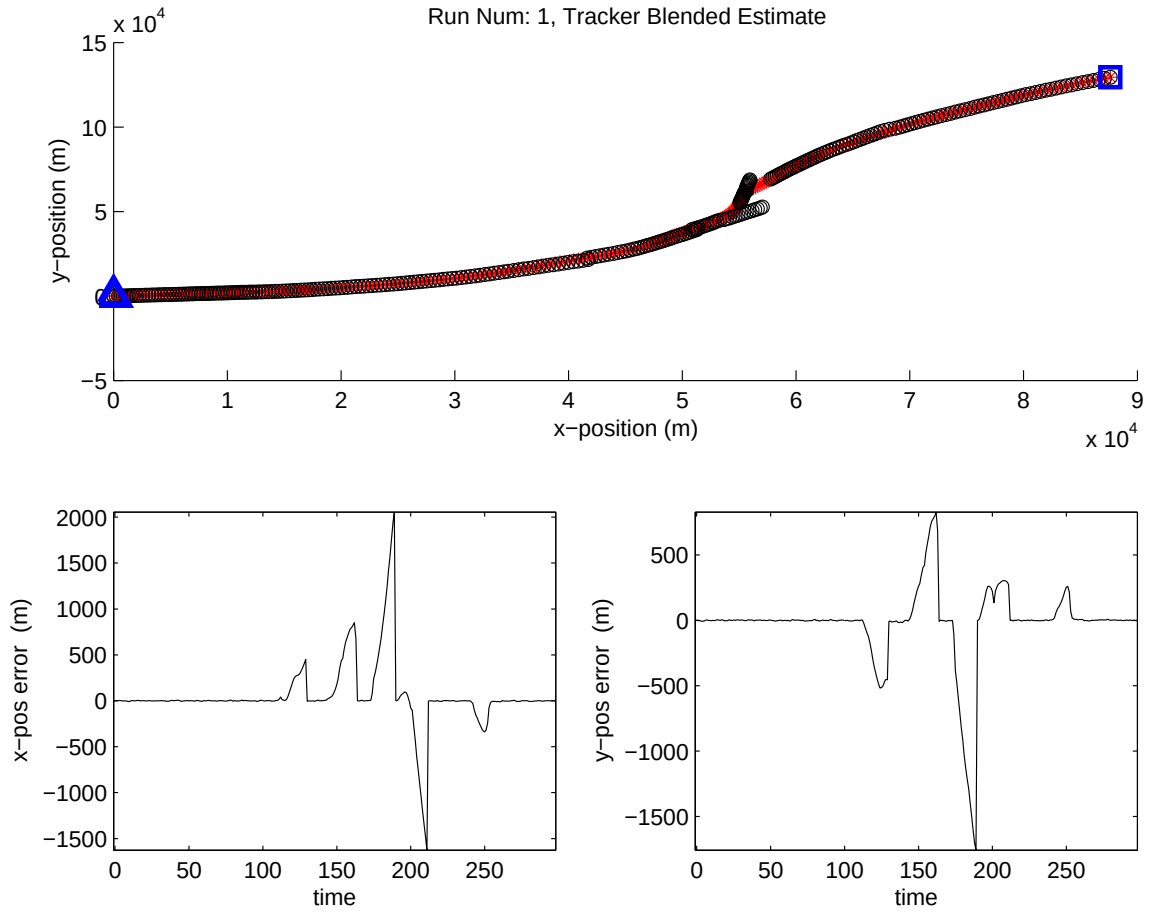

Figure 4.62: Blended output on run number 1. (Suite 4 vs. Scenario 4: MMAE)

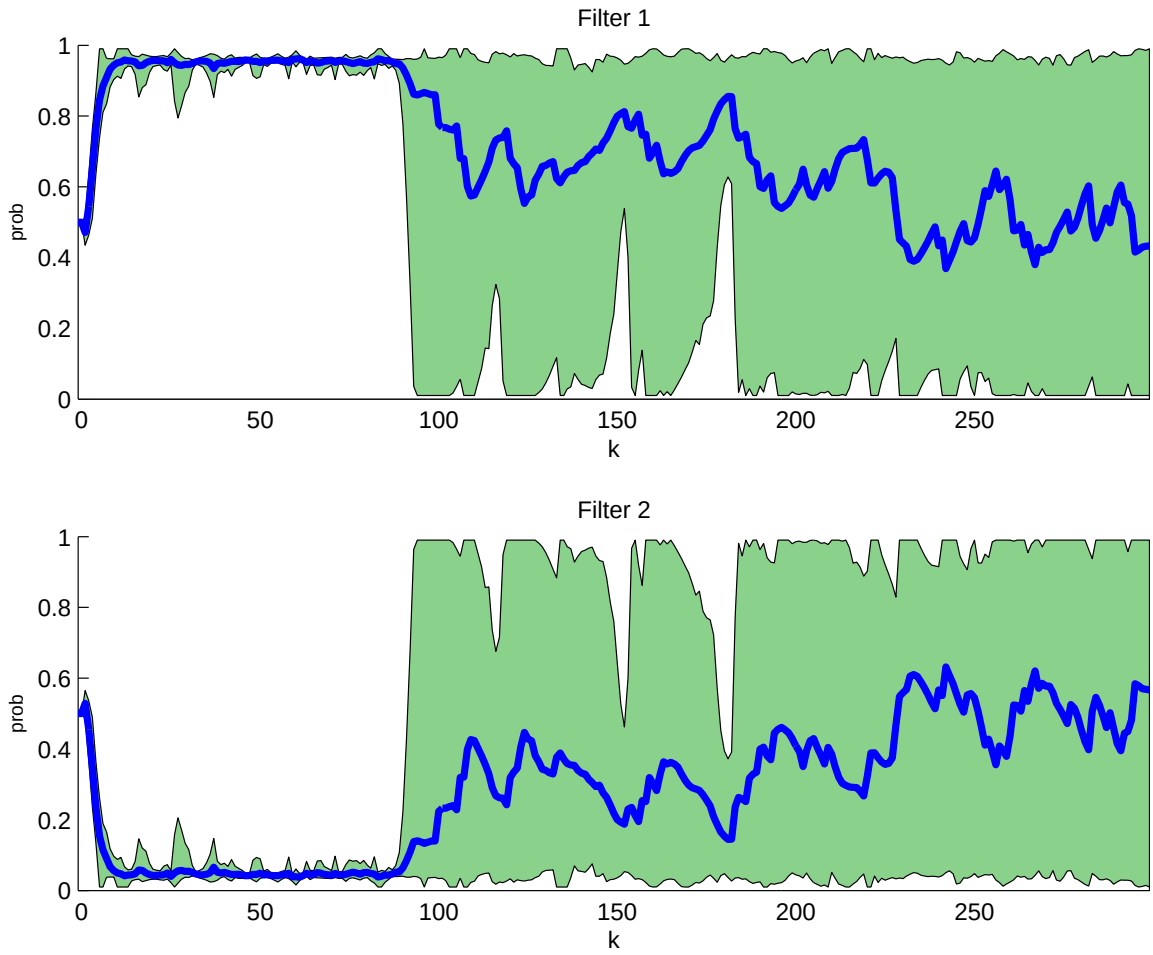

Figure 4.63: Probability flow. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

The IMM with a transition-resistant Markov matrix performed quite differently. Notice the diluted probability flow in Fig. 4.63 compared to that seen in Fig. 4.59, and notice how the benign filter never retakes the probability weight even though the benign model is again in effect after ($k = 200$). This behavior makes sense, because in IMM, the estimates provided by elemental filters 1 and 2 are *no longer* associated with static declarations of an appropriate dynamics model. The intermixing of estimates among all elemental filters at every algorithm cycle cause the state estimate in any given elemental filter to be the product of a *history* of dynamics models in force, whereas in MMAE, the estimate in any elemental filter is the product of that filter's dynamics model *only* (so long as the filter of interest has not undergone a divergent filter reset in recent cycles).

In Fig. 4.64, it is clear the blended estimate is diverging during one or more of the runs, and on some of these runs, the tracker did not recover. Compare this to the MMAE results of Fig. 4.58,

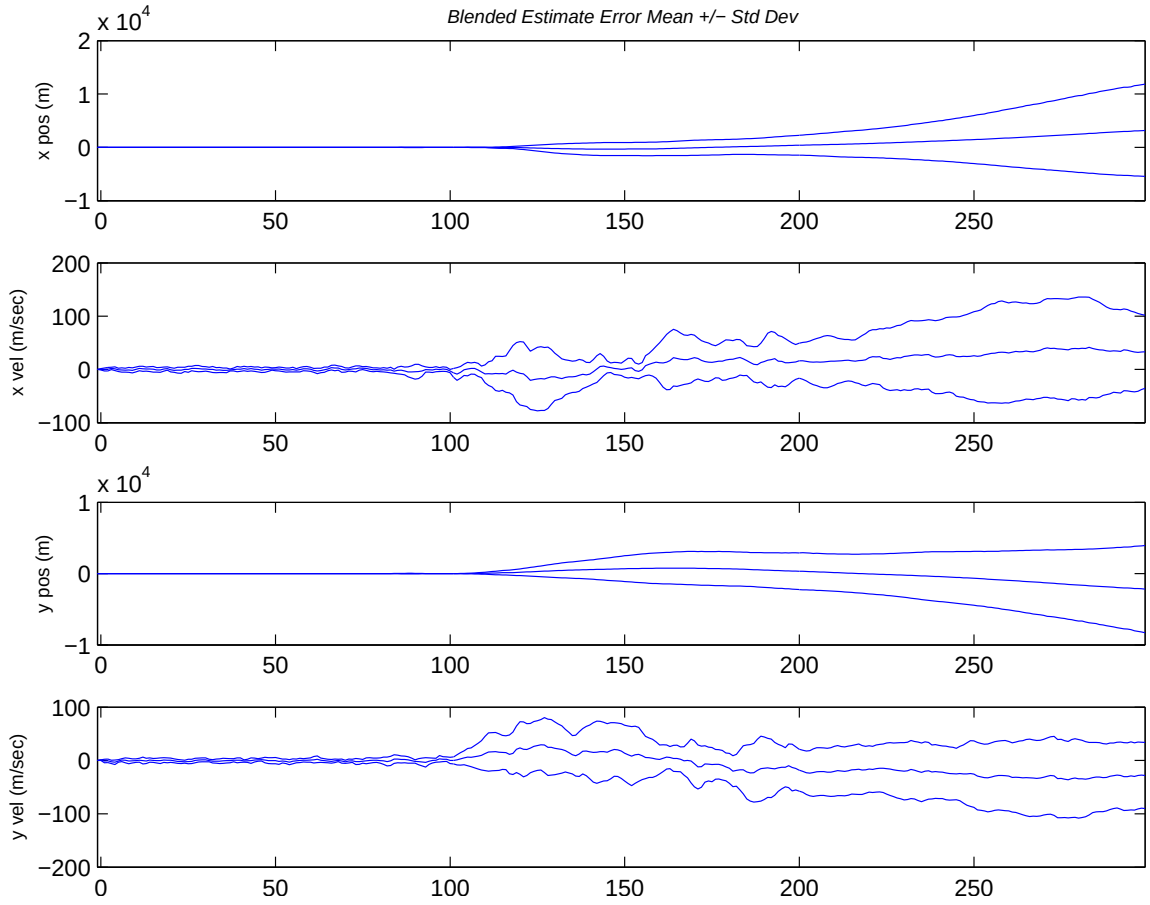

Figure 4.64: Blended estimate. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

in which the very tight mean $\pm 1\sigma$ error plots during the final phases of flight (after approximately ($k = 250$)) indicate the tracker recovered on all runs. On the other hand, in Monte Carlo run 1, the discontinuities present in the MMAE solution are not present in the IMM (see Figs. 4.65, 4.66, and 4.67). Note the trajectories give the appearance that both IMM elemental filters have identical estimates, but this is merely a result of plot scaling. The filter state estimates $\hat{\boldsymbol{x}}(k|k)$ were verified to be unique by looking at the tracker output data files.

Except for the error spikes appearing around ($k = 275$) in both axes, the IMM filter errors and blended estimate errors are very small. Whereas the equivalent MMAE results in Figs. 4.60 through 4.62 show multiple spikes of varying magnitude (sometimes as large as 2 km in a single axis), the IMM only exhibits this single spike in each axis. The specific cause of the spike is unknown, but as seen in most of the MHT algorithm results, even filter configurations that are well-matched to the truth model may yield state estimates that jump as the MHT refines its Gaussian mixture to best reflect the assumed target trajectory. Furthermore, the IMM and the MMAE, even with identical elemental filter configurations, will gate measurements according to the mixture component estimates contained within each filter (which are generally different for an MMAE versus an IMM). It is conceivable that, at the single time step associated with the spike, an atypical measurement gate situation occurred, momentarily disrupting the IMM.

Further investigation showed the IMM with the transition-resistant Markov matrix suffered total track loss on several runs, resulting in the divergent errors in the blended estimate summary. This is evidenced in Fig. 4.64, in which mean $\pm 1\sigma$ lines of the the actual error committed grow unbounded as time progresses. Conversely, the MMAE always recovered from momentary track loss, evidenced by the fact that short-duration excursions in the error plots (caused by momentary loss of track) always snap back to a tracking solution (shown in Fig. 4.58). The only explanation (discernible by this author) that might account for the difference in track performance is that the equal mixing of states (as evidenced by the diluted probability flows) makes *neither* filter perform well enough to maintain track. On the other hand, MMAE allows each filter to run uninterrupted until it is declared divergent, so a well-performing filter is not forced to take a poorly suited initial condition at the beginning of every cycle.

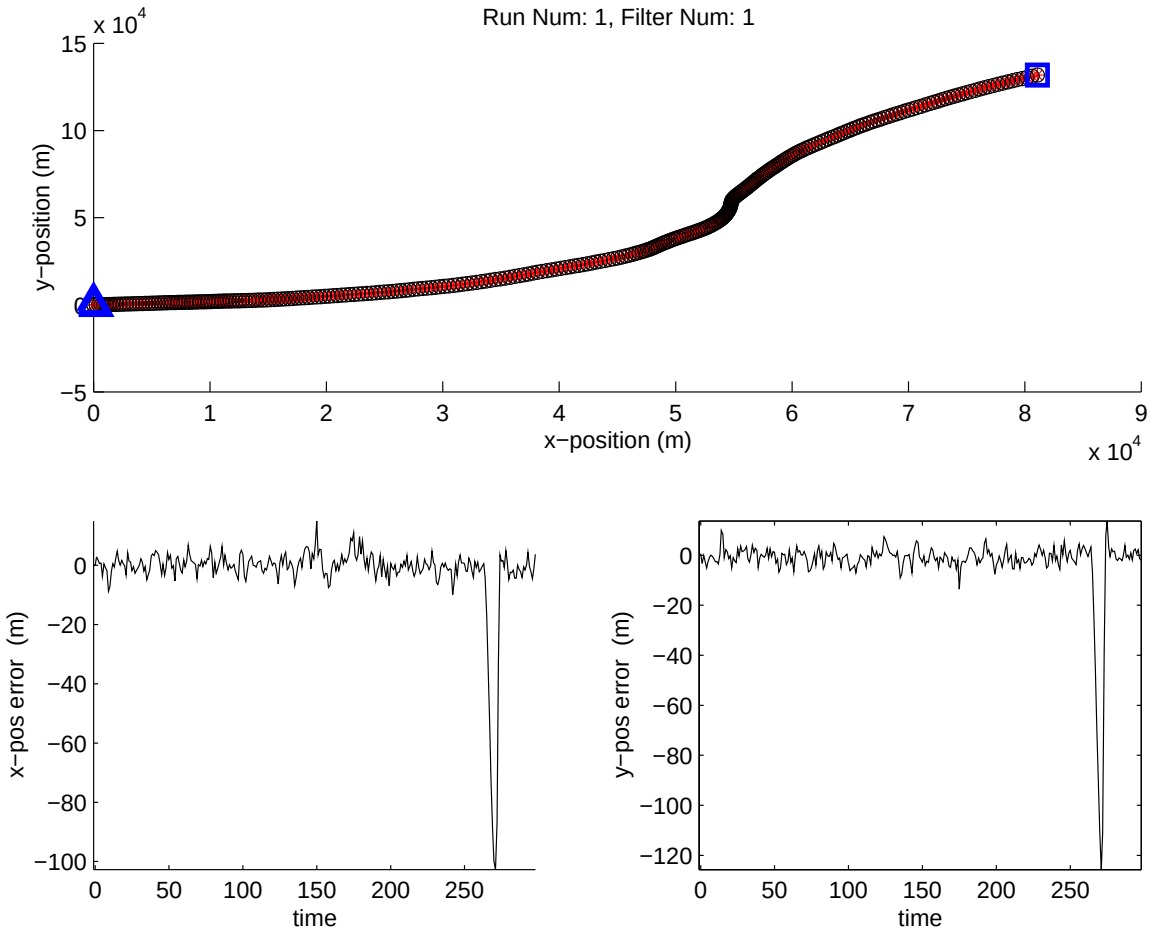

Figure 4.65: Filter 1 estimate. This represents a "best case" run of the 10 trials conducted. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

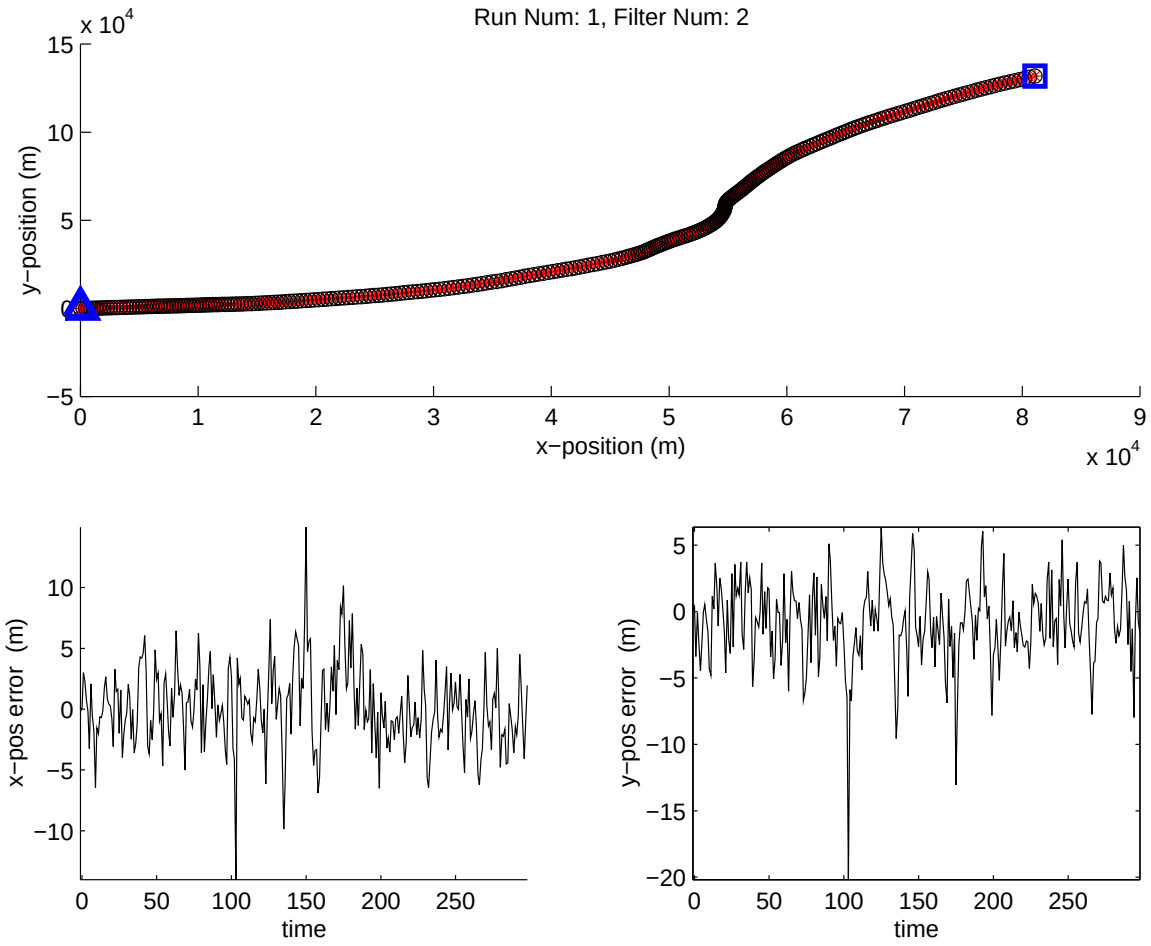

Figure 4.66: Filter 2 estimate. This represents a "best case" run of the 10 trials conducted. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

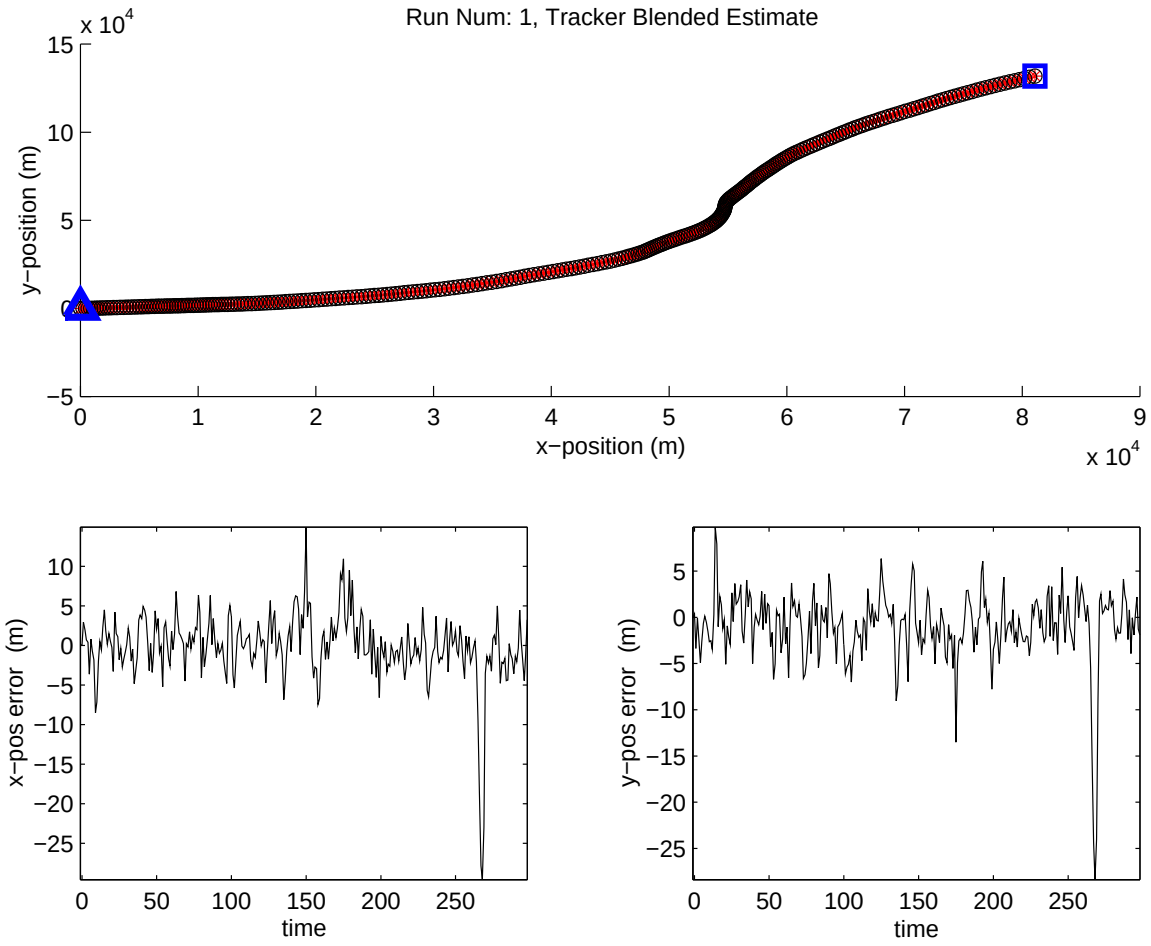

Figure 4.67: Blended estimate. This represents a "best case" run of the 10 trials conducted. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

4.4.4 Configurations with Two TPV Filters and One Constant-Velocity or One FOGMA Filter. The Kalman-filter-based performance tests showed that TPV filters were essential to tracking a target executing TPV maneuvers, so single-filter suites were not tested against TPV scenarios under the measurement association uncertainty cases.

The first cases to be considered will be those with constant-velocity dynamics during benign phases of flight. Per the discussions in Section 4.1.6, these truth scenarios, which include only 4-state dynamics models, provide the closest match to the TPV filters used in this research.

Figures 4.68 through 4.71 show the performance of an MMAE with filter Suite 5 running against truth Scenario 5 on Monte Carlo trial number 1. One of the most interesting aspects of the case here is that filter 2, the -3-g TPV filter, has almost zero state estimate error for a large portion of the simulation. Filter 3, the CV filter, has zero error until the onset of the first TPV maneuver at ($k = 40$) and begins a return to essentially zero error after the conclusion of the final TPV maneuver at ($k = 200$). This shows that the elemental filters themselves appear to be doing a reasonably good job of tracking the truth trajectory. The blended estimate, unfortunately, does not adequately represent the best estimate provided by the filter bank at any given time. In fact, the blended estimate error is rather noisy compared to that of the elemental filters taken individually.

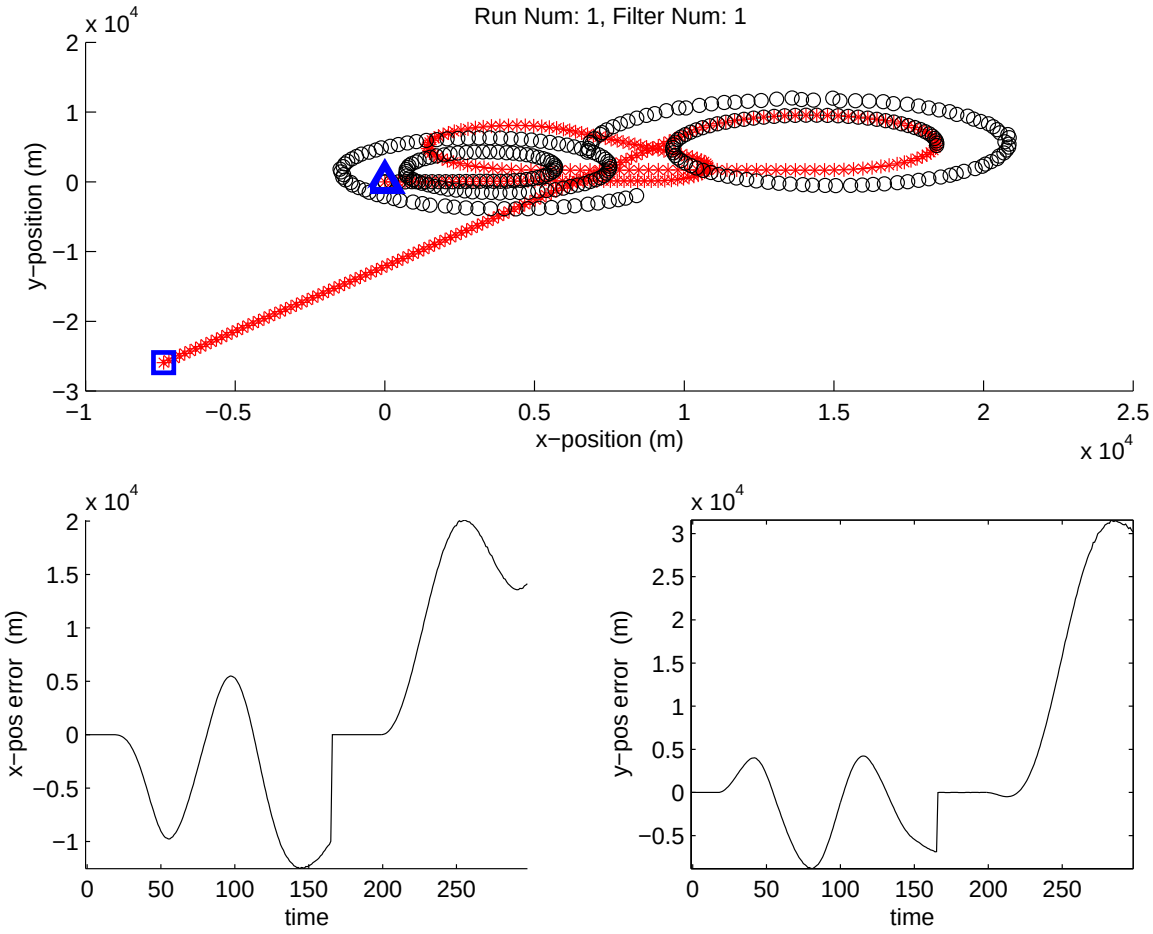

Figure 4.68: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

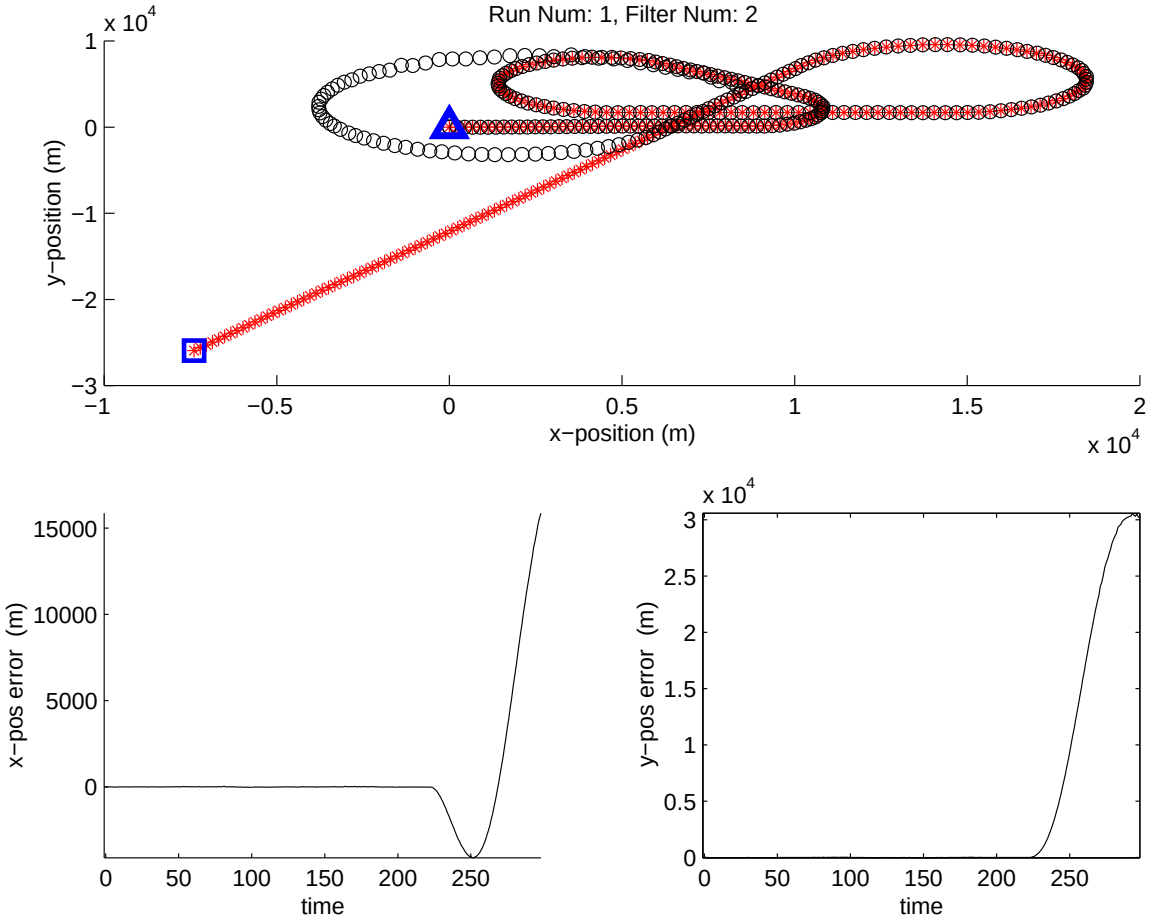

Figure 4.69: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

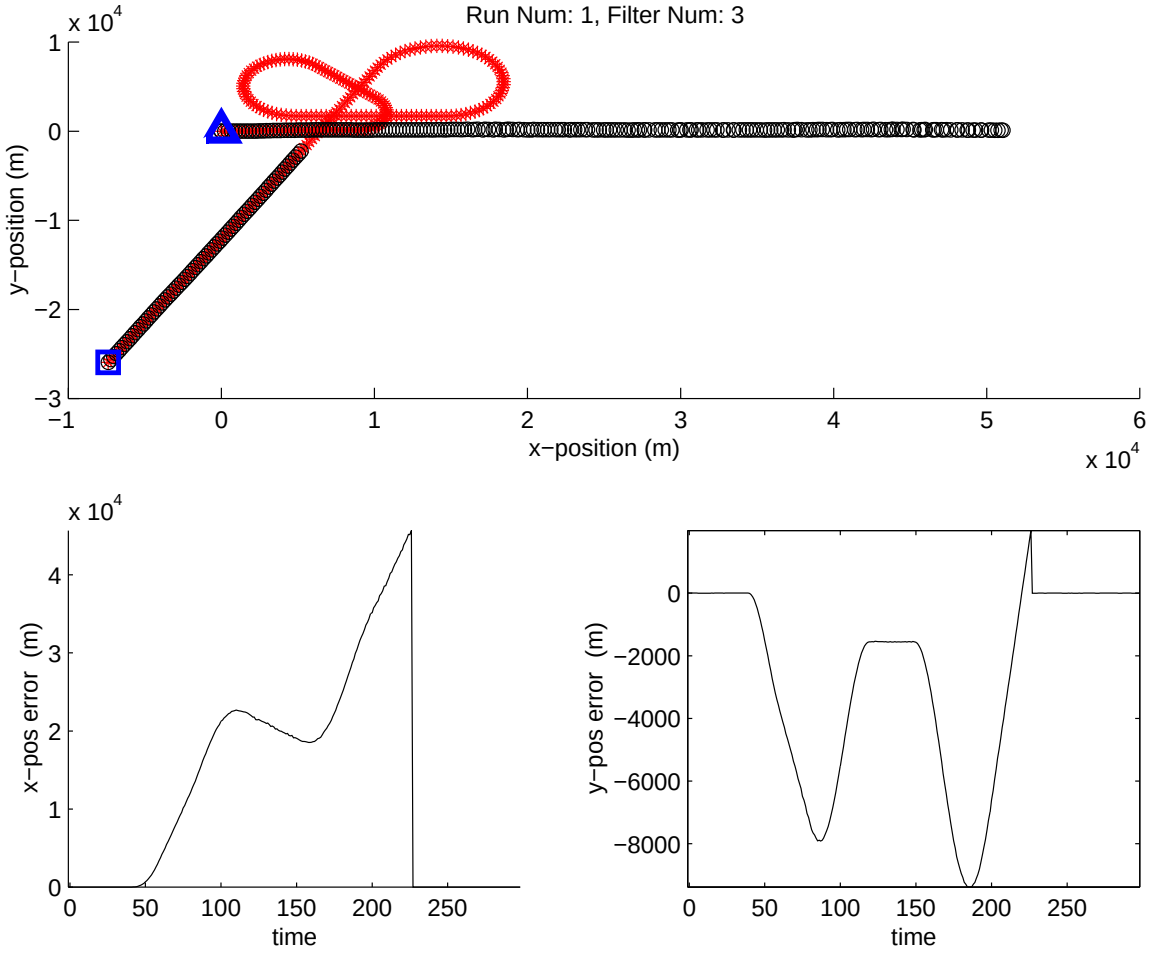

Figure 4.70: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

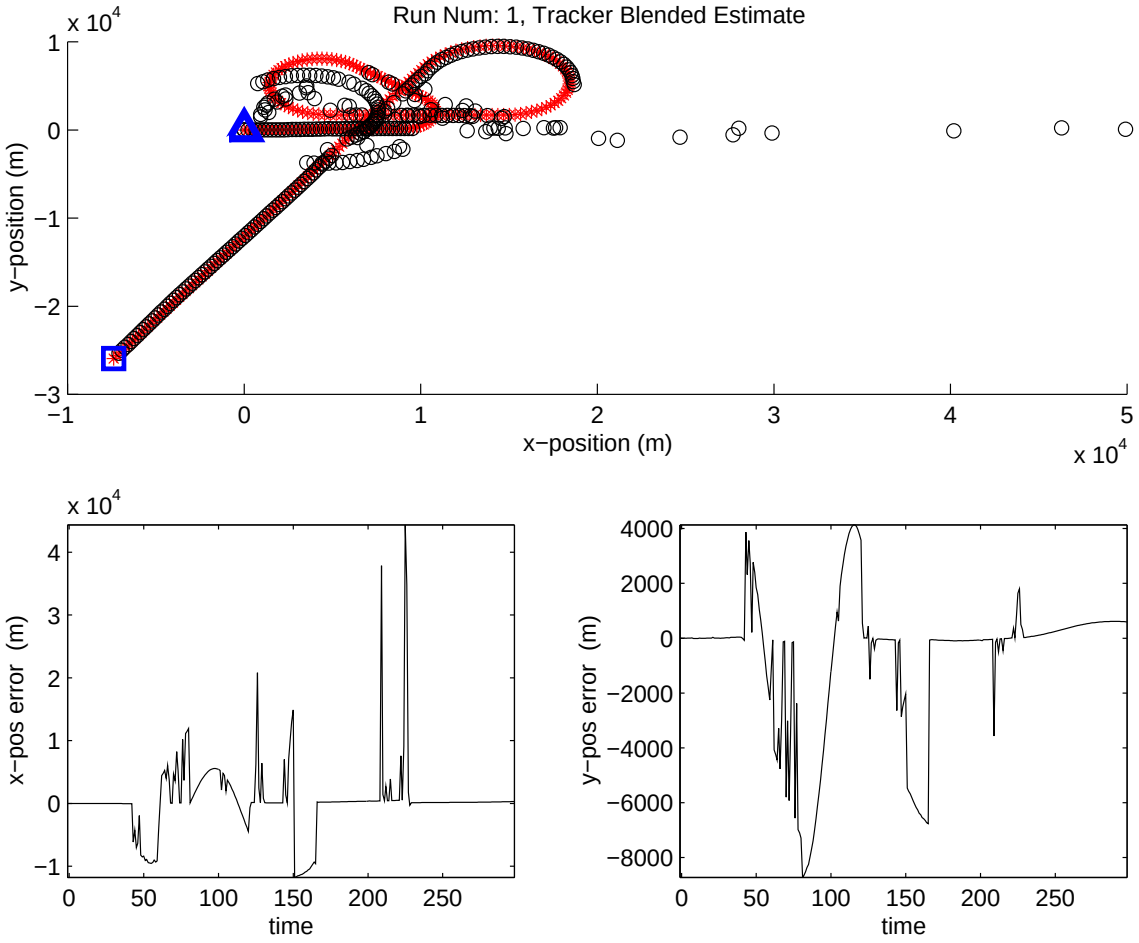

Figure 4.71: Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

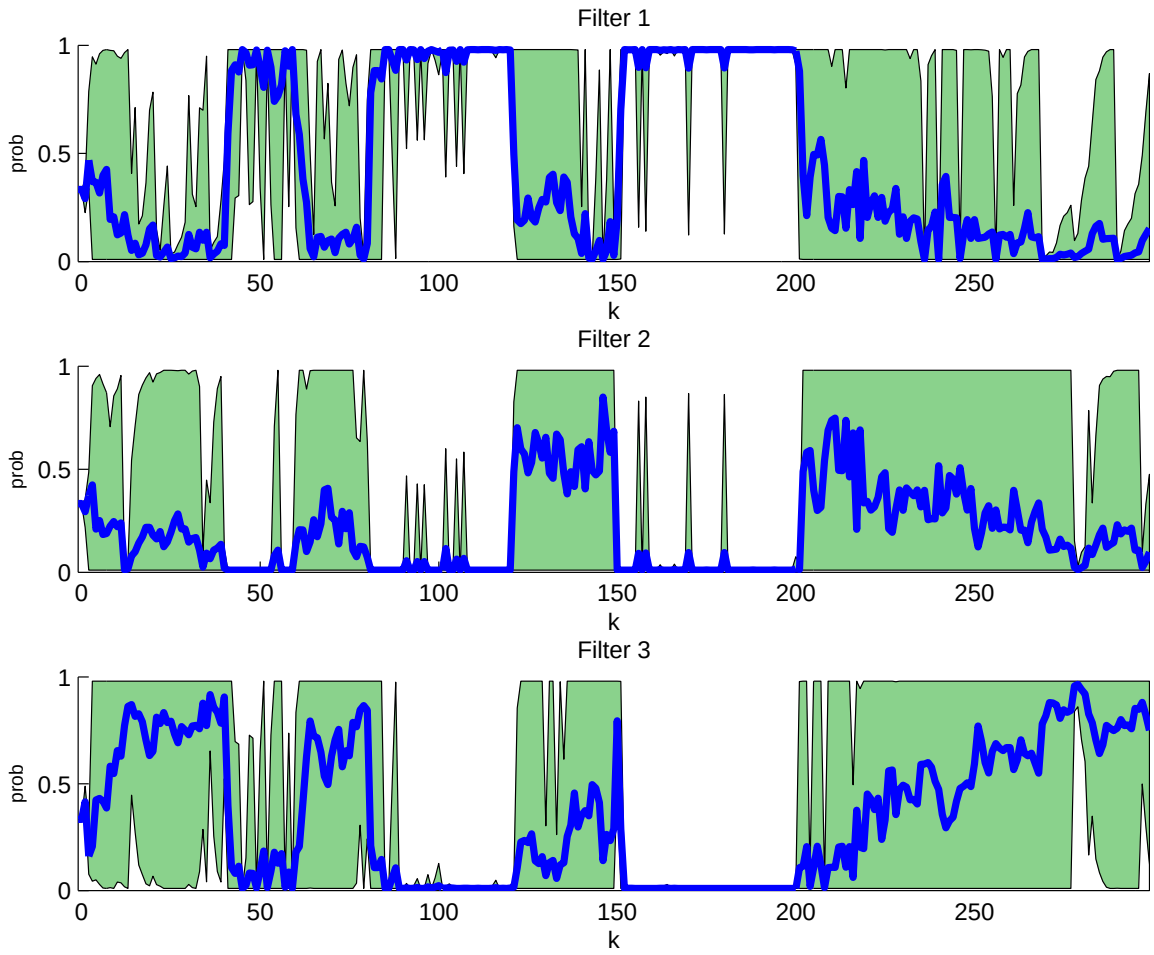

Figure 4.72: Probability flow for 10 Monte Carlo runs. Filter 1 should be the maximally weighted filter (Suite 5 vs. Scenario 5: MMAE)

Of more interest are the summary plots, shown in Figs. 4.72 through 4.76 for the 10 Monte Carlo runs. For comparison, the no-clutter equivalent results appear in Figs. 4.16 through 4.20. As a reminder, the +3-g TPV maneuvers occur between $(k = 40, \ldots, 60)$, $(k = 80, \ldots, 120)$, and $(k = 150, \ldots, 200)$. First, notice in Fig. 4.72 how filter 1 consistently takes the probability weight during the phases of filter-matched TPV maneuvers, as expected. Filter 3 does a reasonably good job of regaining filter probability weight between maneuvers, and clearly gains weight once the maneuvers are complete at $(k = 200)$. Also note that filter 1's $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ (seen in Fig. 4.73) becomes very small during these maneuvers and larger elsewhere, similar to, but not exactly like the Kalman-filter-based case (see Fig. 4.17). Finally, during the phases in which maneuvers are occurring, the blended solution is maintaining track and quickly recovers once all maneuvers are finished.

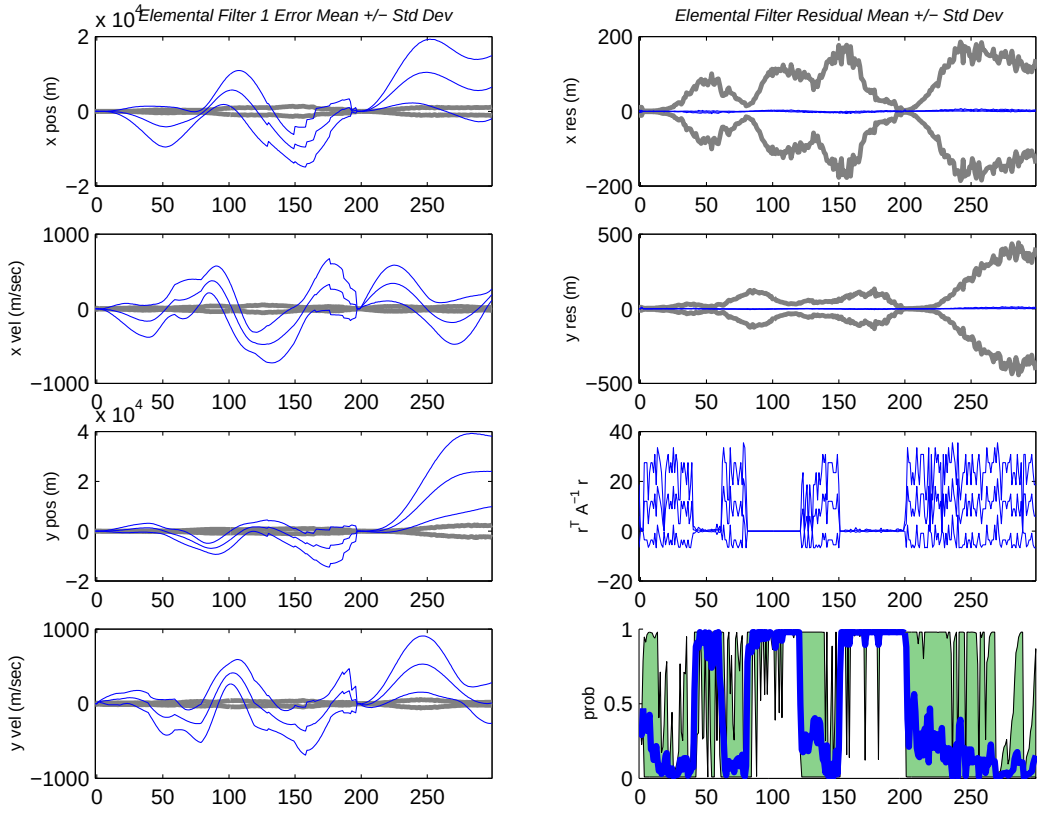

Figure 4.73: Filter 1 data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

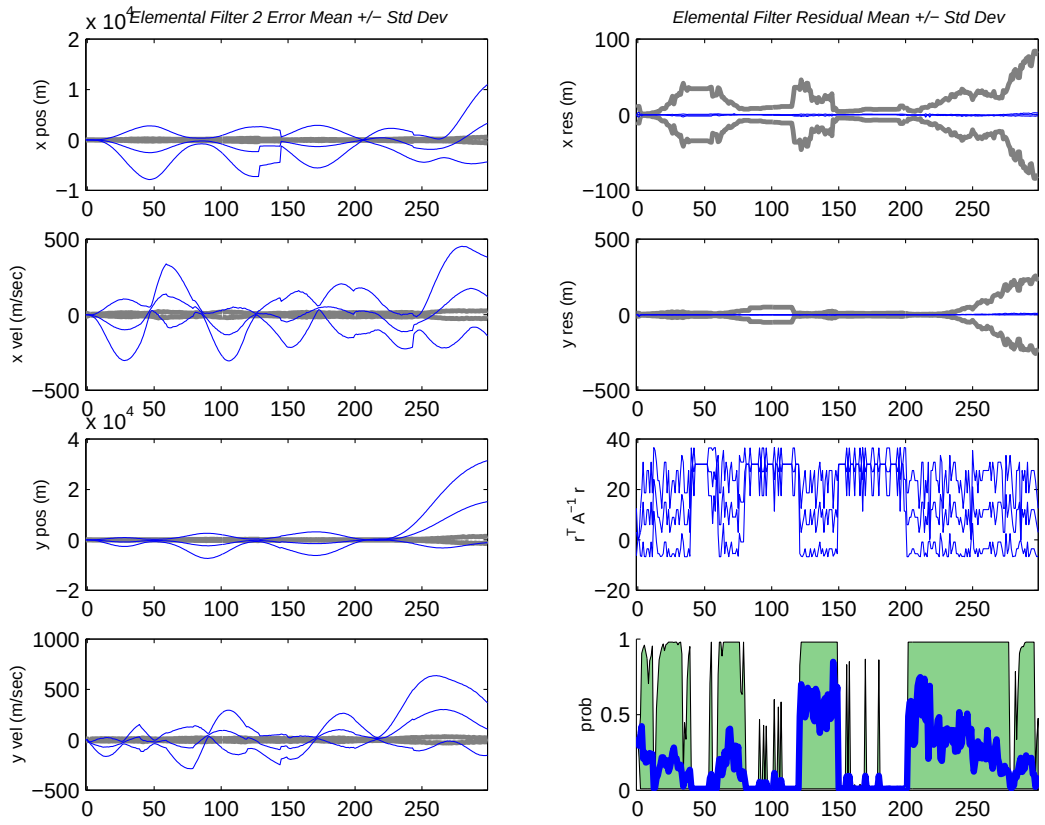

Figure 4.74: Filter 2 data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

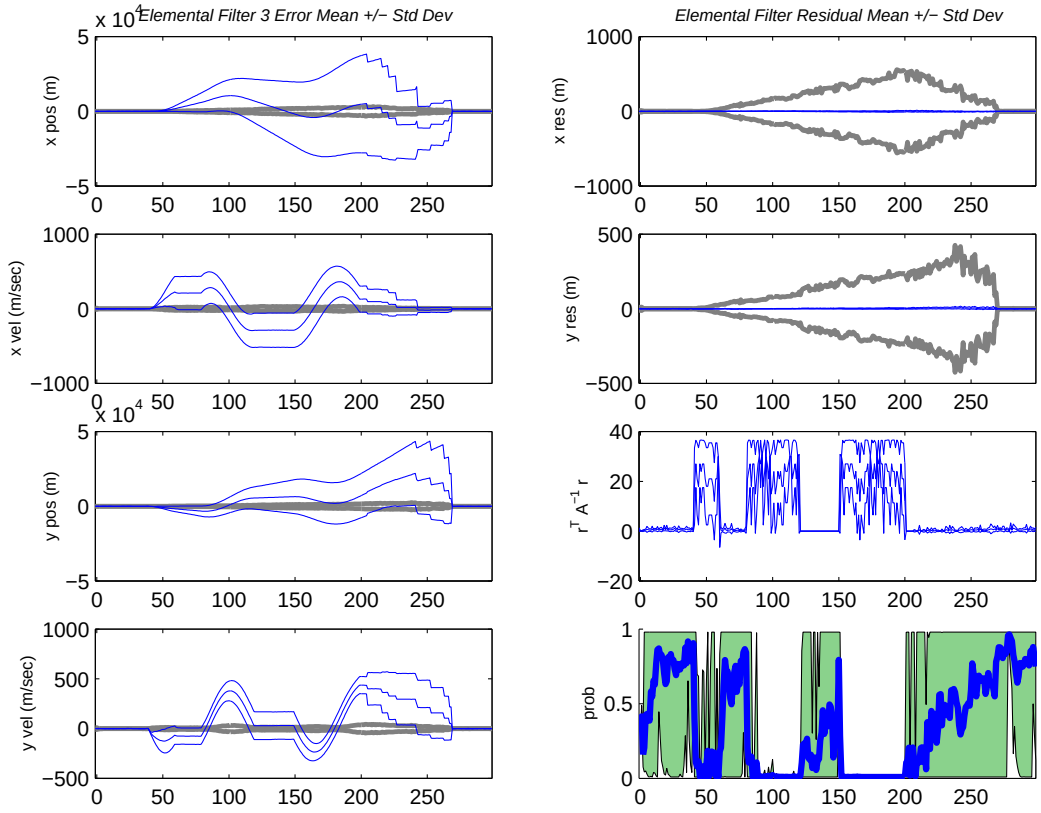

Figure 4.75: Filter 3 data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

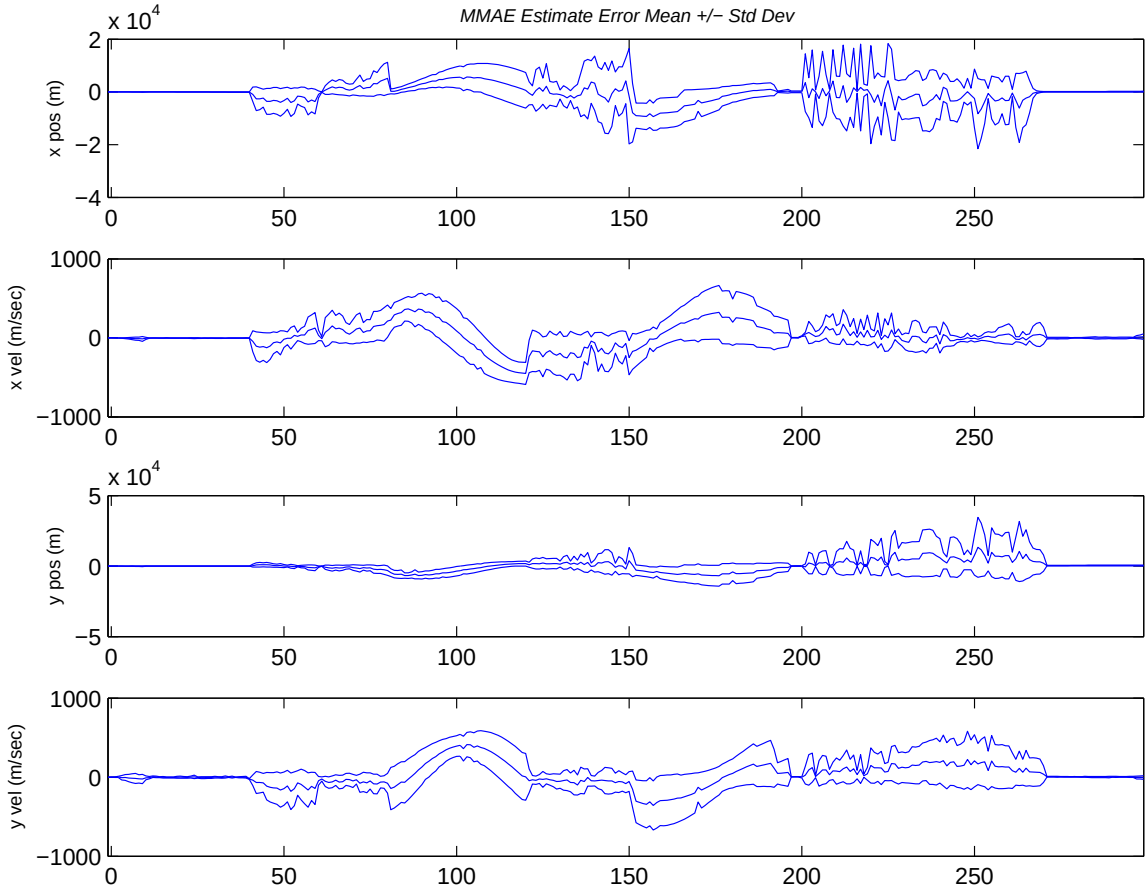

Figure 4.76: Blended estimate data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

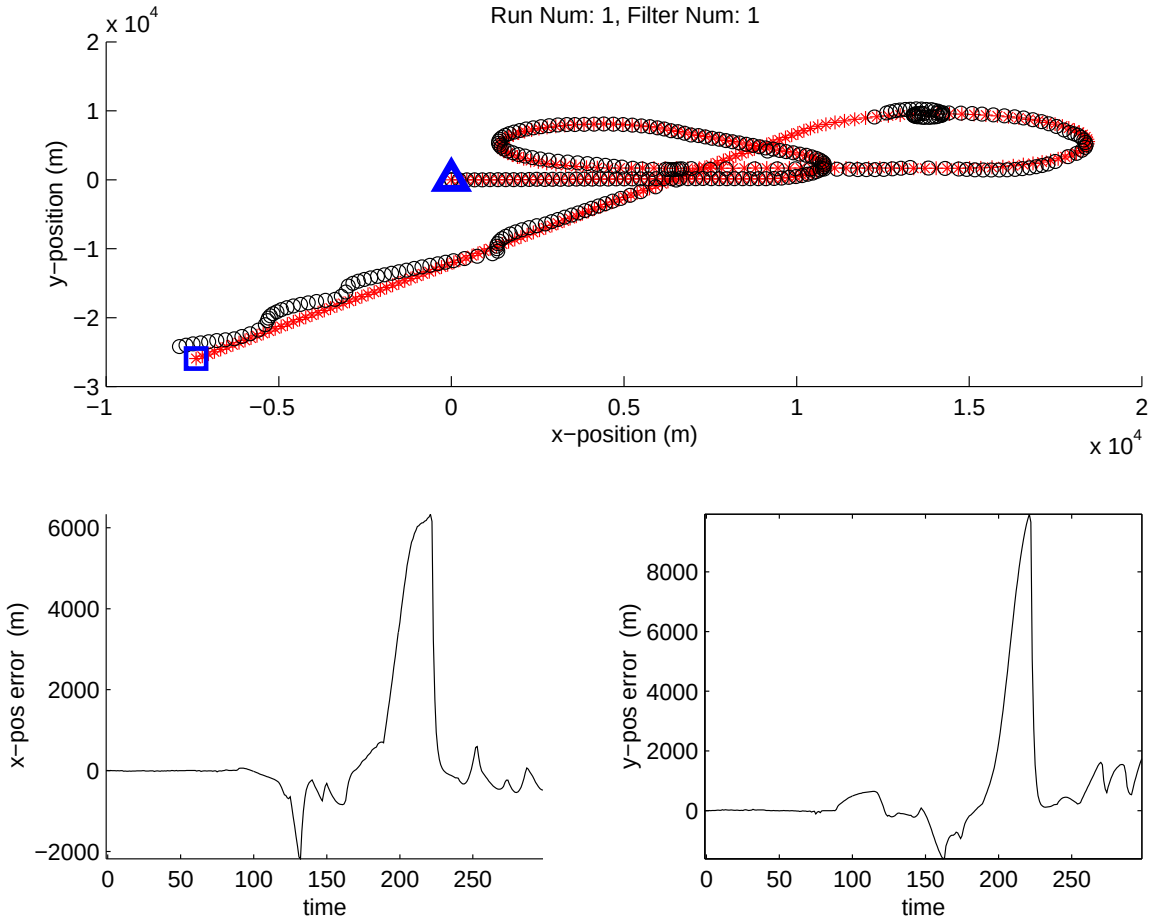

Figure 4.77: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

The IMM case with a non-transition-favoring Markov matrix appeared to perform slightly better than the MMAE on Monte Carlo run 1. It had a blended estimate that was less noisy (Fig. 4.80) and elemental filter estimates with consistently smaller errors (Figs. 4.77 through 4.79). A comparison between Fig. 4.71 and Fig. 4.80 show clear differences in behavior under identical truth target trajectories and identical measurement sets (i.e., on the same Monte Carlo trial, the truth state and *all* measurements in the simulation are the identical in both configurations). The error spike around ($k = 200$) is probably an artifact of the measurement gating environment, and it notably occurs at the cessation of TPV maneuvers (the last of which occurs at ($k = 200$)). Presumably, it took several sample periods for the elemental filters and blended estimate to recover from the change in truth model dynamics. Note that different plot scalings for the MMAE and IMM may give the appearance that the two trials are not the same, but they are indeed identical.

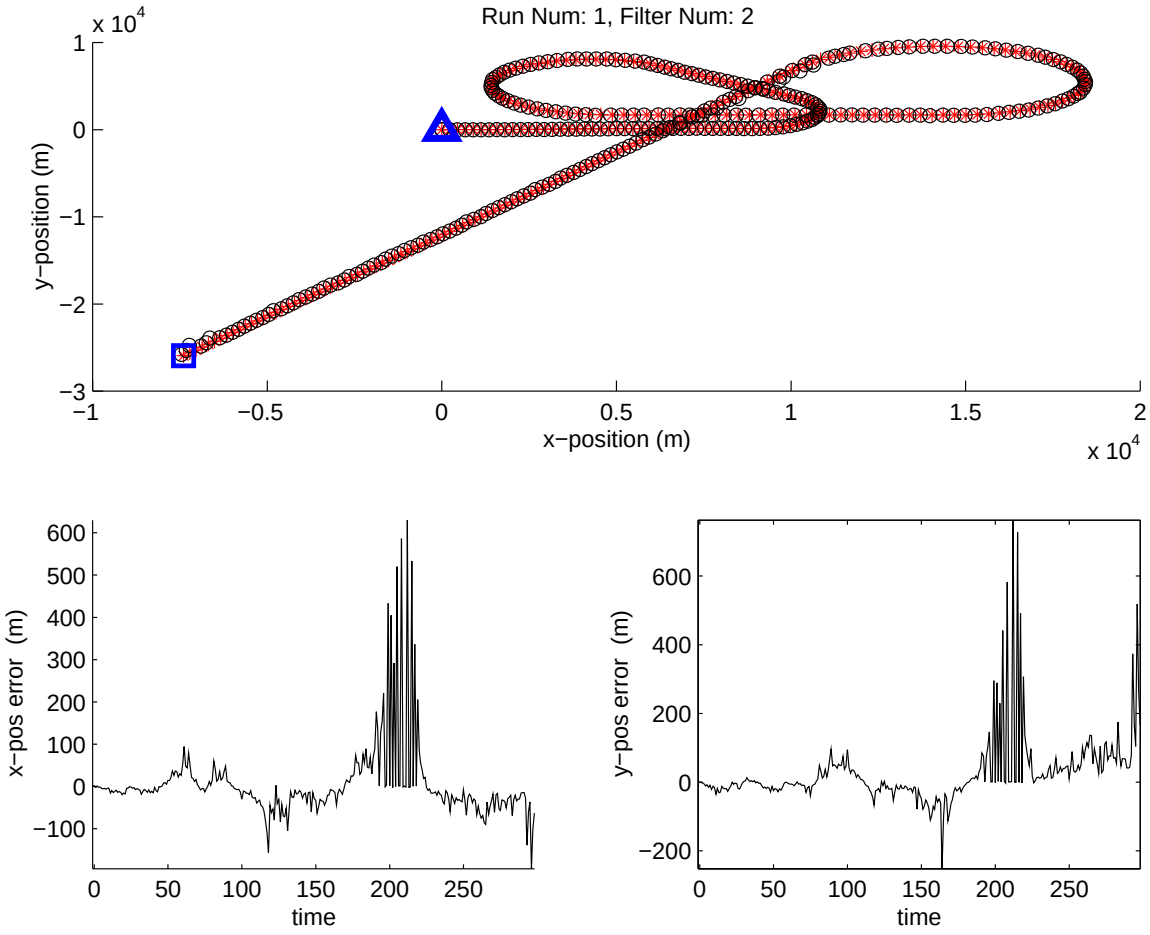

Figure 4.78: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

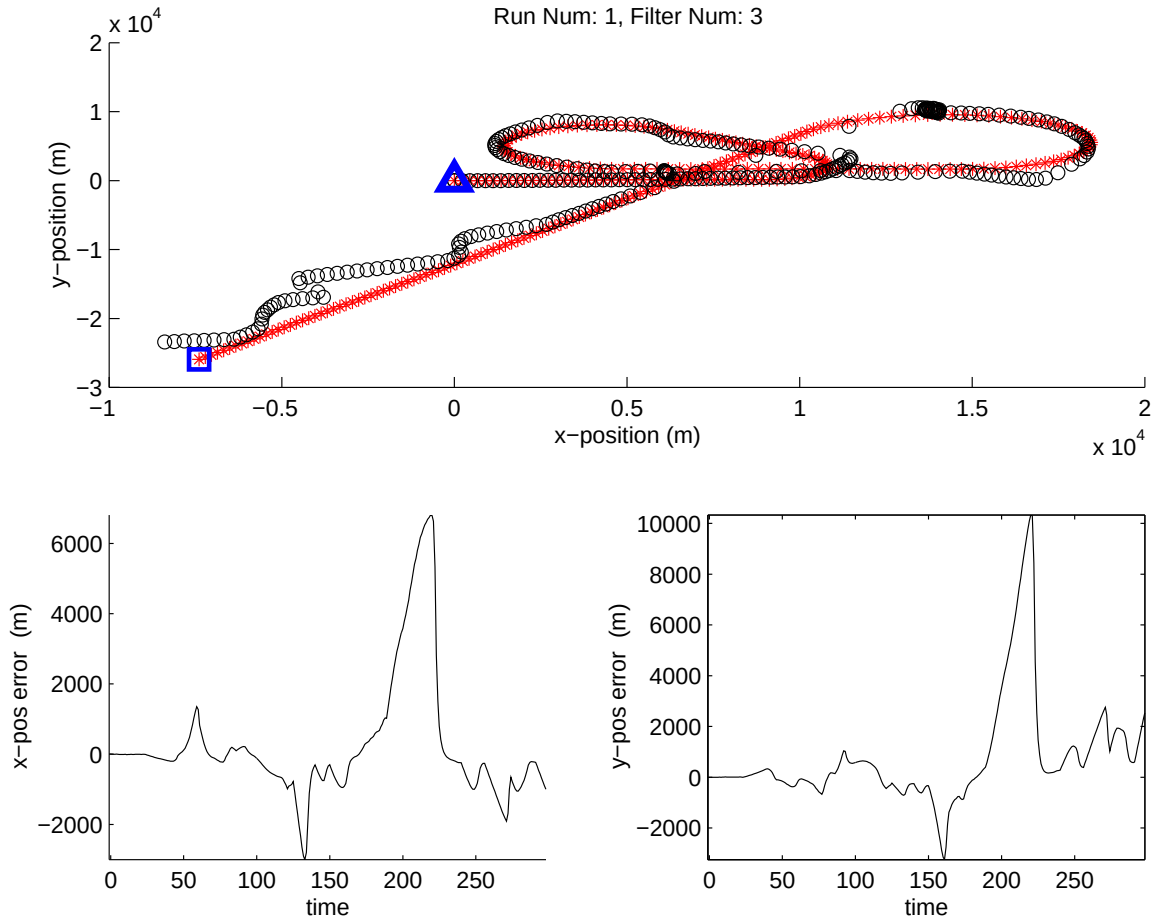

Figure 4.79: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

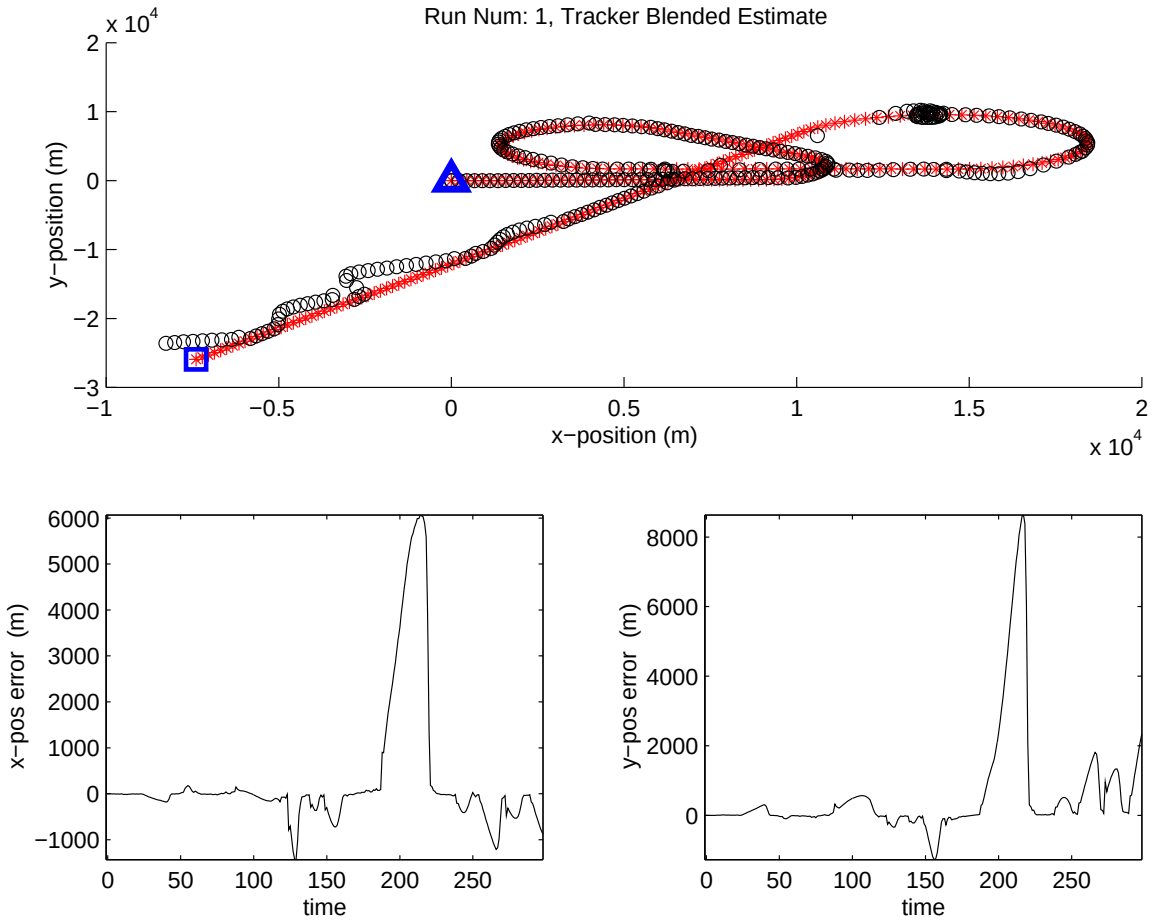

Figure 4.80: IMM Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

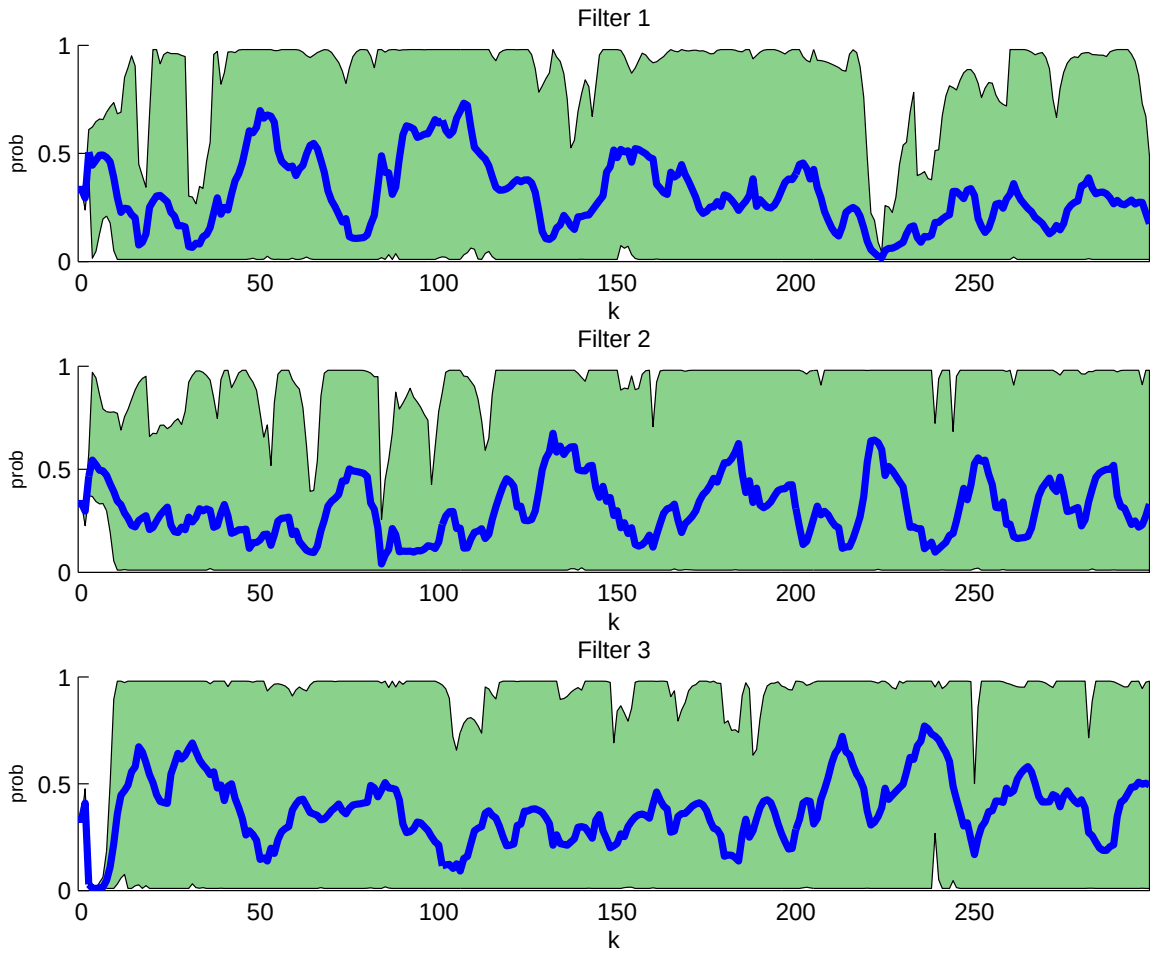

Figure 4.81: Probability flow for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

Of particular interest, again, is the IMM's very ambiguous probability flows (Fig. 4.81) compared to that of the MMAE (Fig. 4.72). In the IMM, it is essentially impossible to determine what filter most closely matches the truth model dynamics. In the MMAE, there is a very good correlation between the probability flow and the occurrence of TPV maneuvers as defined for truth Scenario 5.

The IMM 10-trial summary plots in Figs. 4.82 through 4.85 exhibit somewhat superior state estimate error performance compared to the MMAE results shown in Figs. 4.73 through 4.76. Looking at the plots in the leftmost column of each figure, notice the difference in scalings between the IMM and MMAE results. One issue of some concern is the differences seen in the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ plots for the MMAE and IMM. The MMAE $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ plots are fairly consistent with what is seen in the equivalent Kalman filter cases (Figs. 4.17 through 4.19), and this explains the

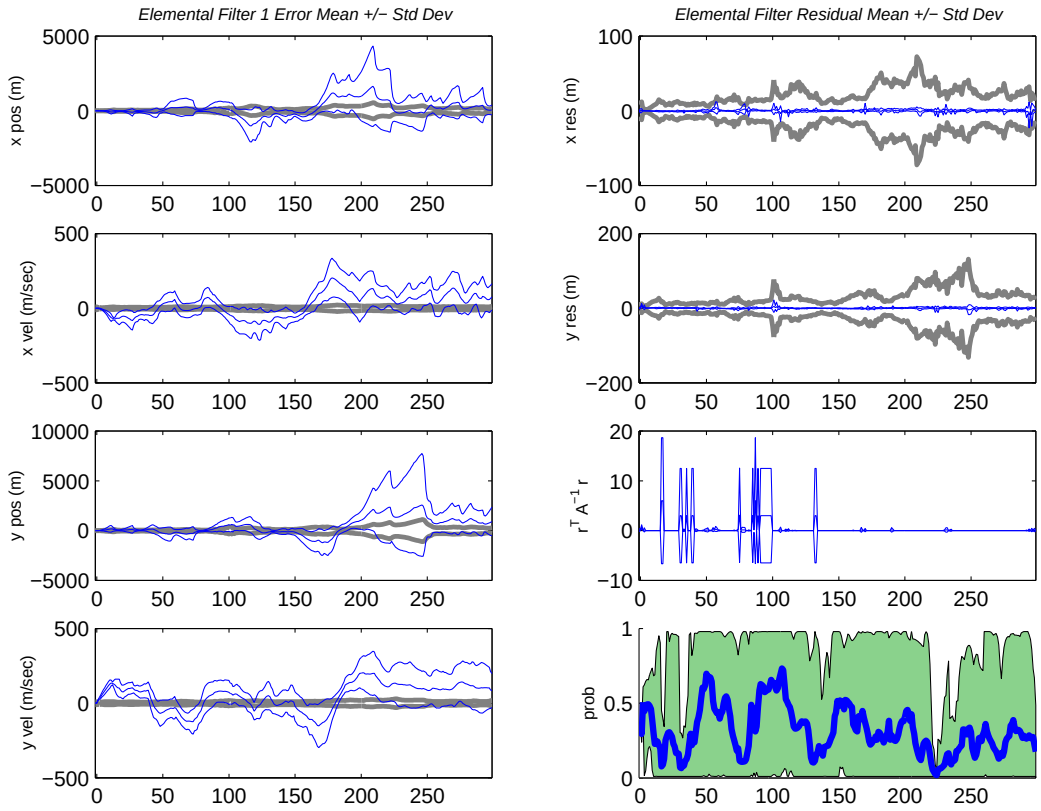

Figure 4.82: Filter 1 data for 10 Monte Carlo runs. (Suite 5vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

similar consistency between probability flows (see Fig. 4.16). The IMM, though, has significantly different $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ plots, even though they *are* consistent with the previously discussed probability flow ambiguity. For example, notice that the values for filters 1 and 3 appear to be almost zero for much of the simulation (note however, that filter 2 does not exhibit the same behavior). The filter-computed pseudo-residual covariance appears to be dominating the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ computation for large portions of the simulation (seen as the wide gray lines in the first and second plots, rightmost column), but the actual residuals shown in these plots are also quite small. In conclusion, it is clear that IMM state intermixing makes filter analysis more difficult. Behavior which might otherwise be construed as a tuning problem in MMAE may, in fact, represent anticipated and acceptable behavior in an identically configured IMM.

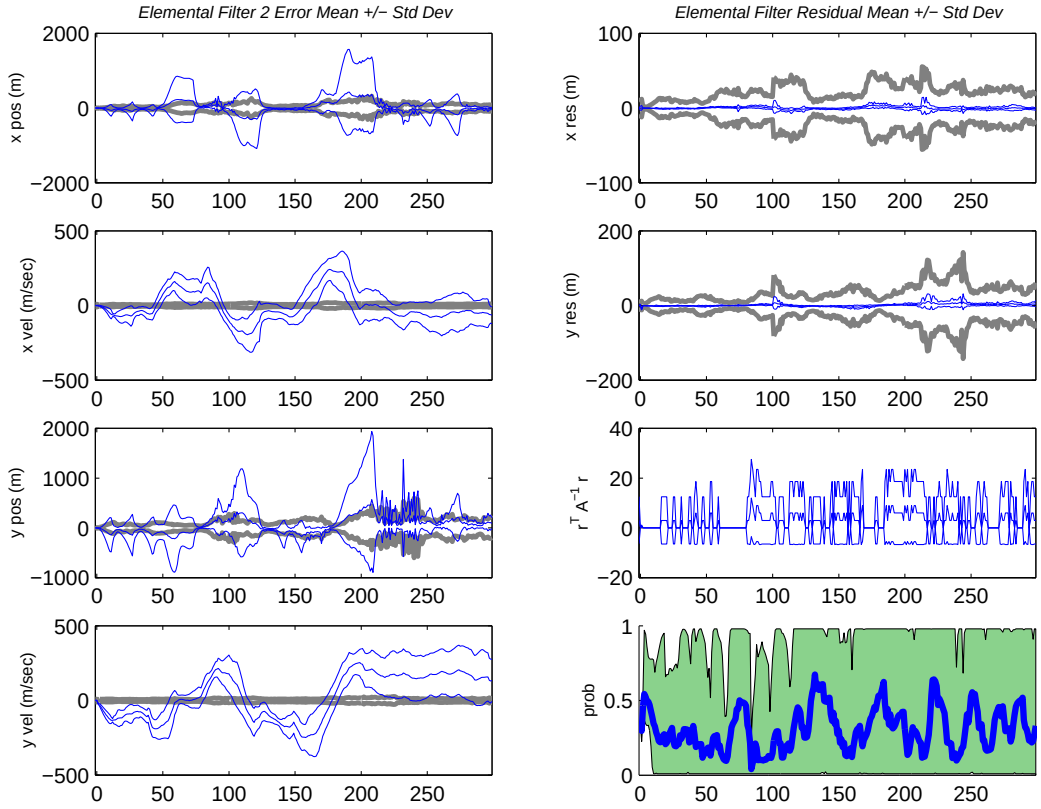

Figure 4.83: Filter 2 data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

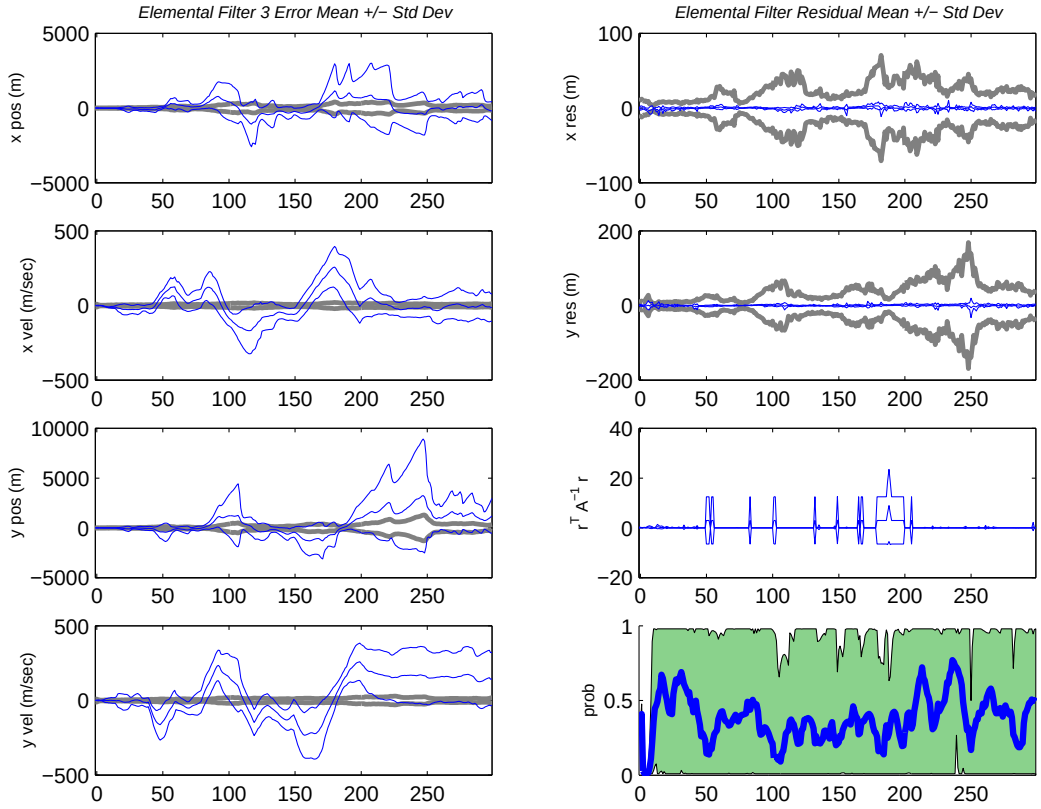

Figure 4.84: Filter 3 data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

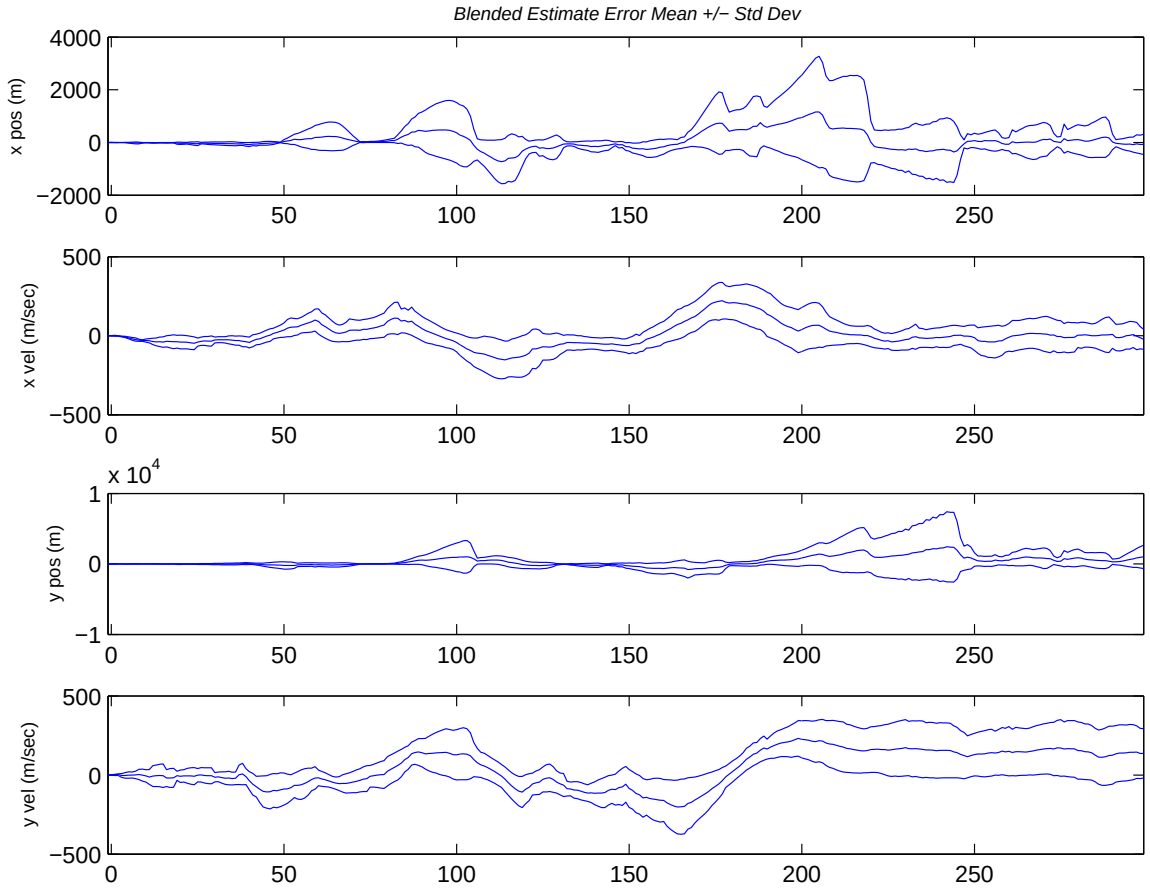

Figure 4.85: IMM Blended estimate data for 10 Monte Carlo runs. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

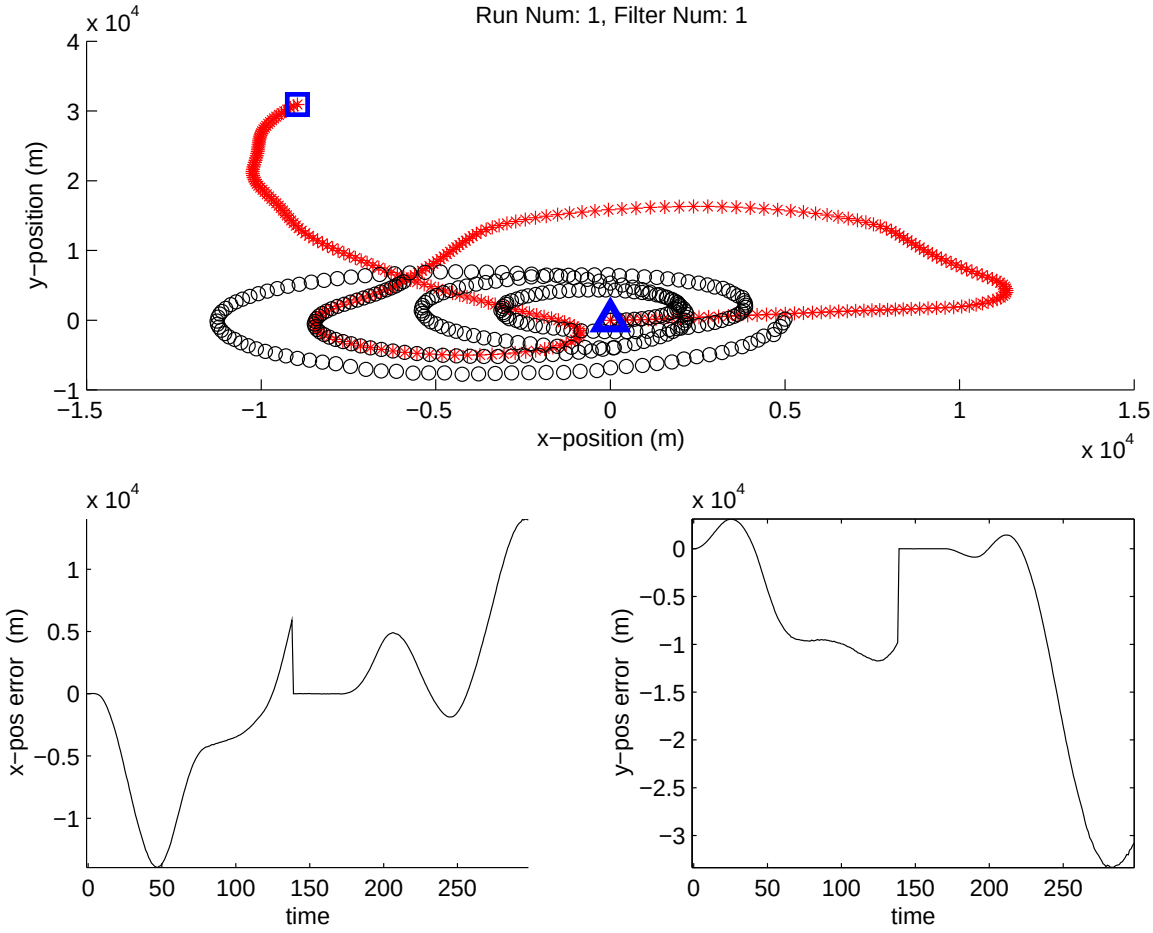

Figure 4.86: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

Suite 6, which has the benign FOGMA filter 3 in place of Suite 5's CV filter 3, performed surprisingly well, despite all of the modelling issues discussed in Section 4.1.6. The MMAE suffered somewhat from the fact that filter resets are delayed by the deferred decision-making properties inherent to an MHT. Monte Carlo run 1 results seemed to be broadly representative of the MMAE's performance under these conditions. Figures 4.86 through 4.89 show the MMAE outputs for the first Monte Carlo run. A best-case run from the set of 10 appeared to be run number 3, which appears in Fig. 4.90.

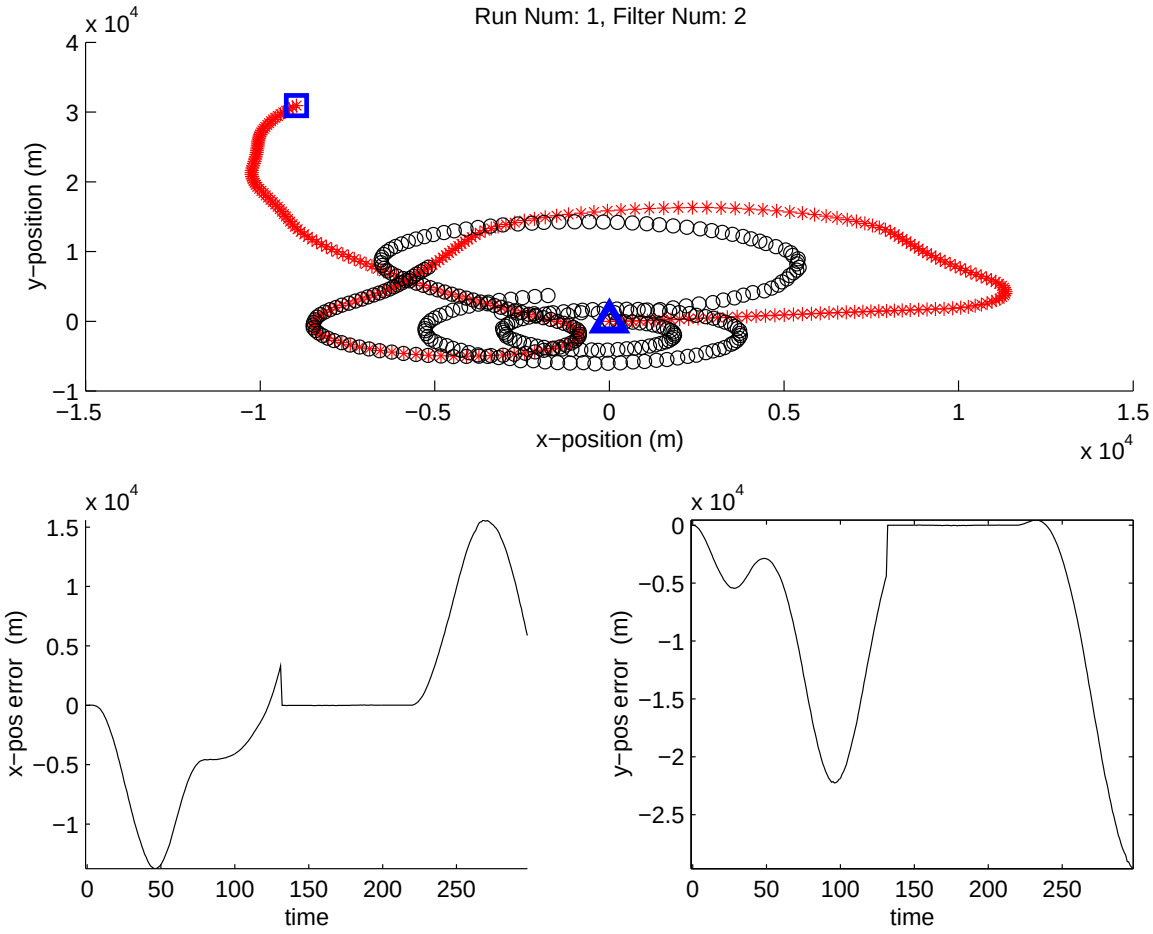

Figure 4.87: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

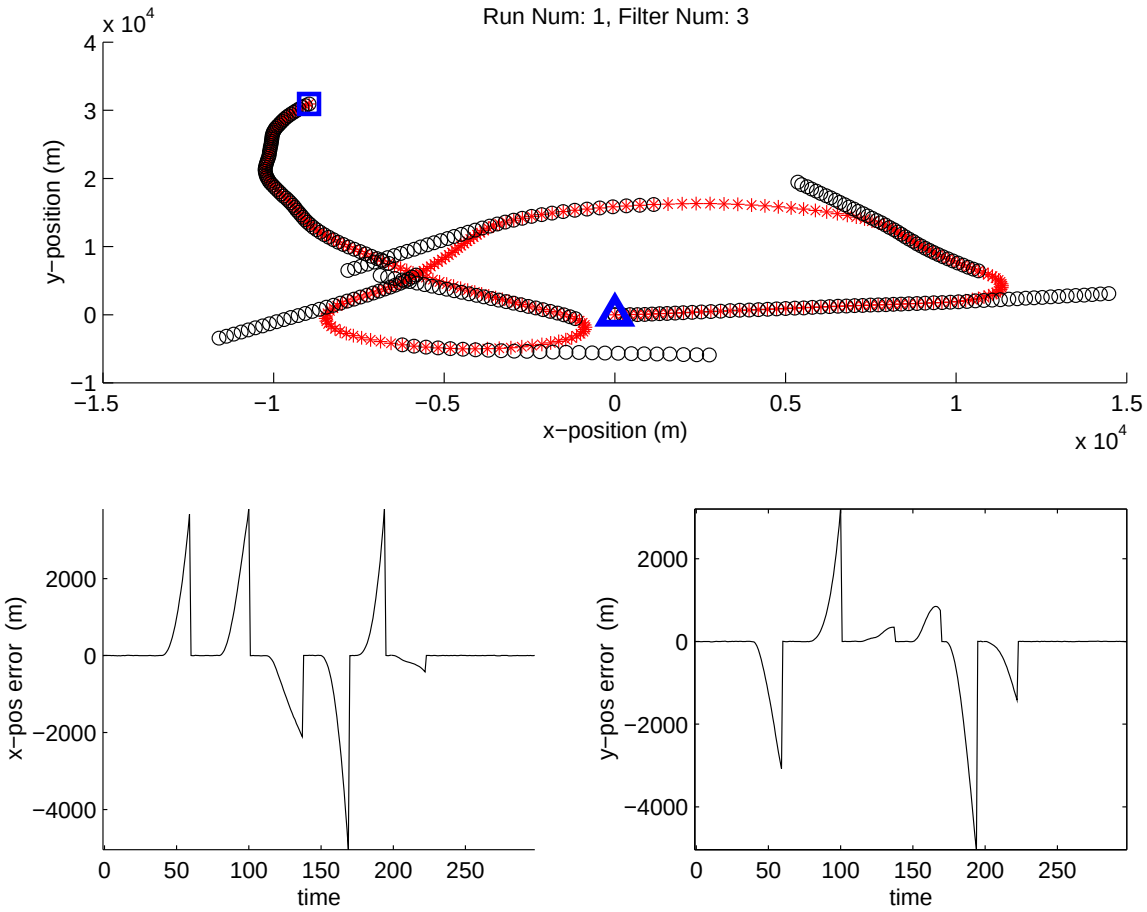

Figure 4.88: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

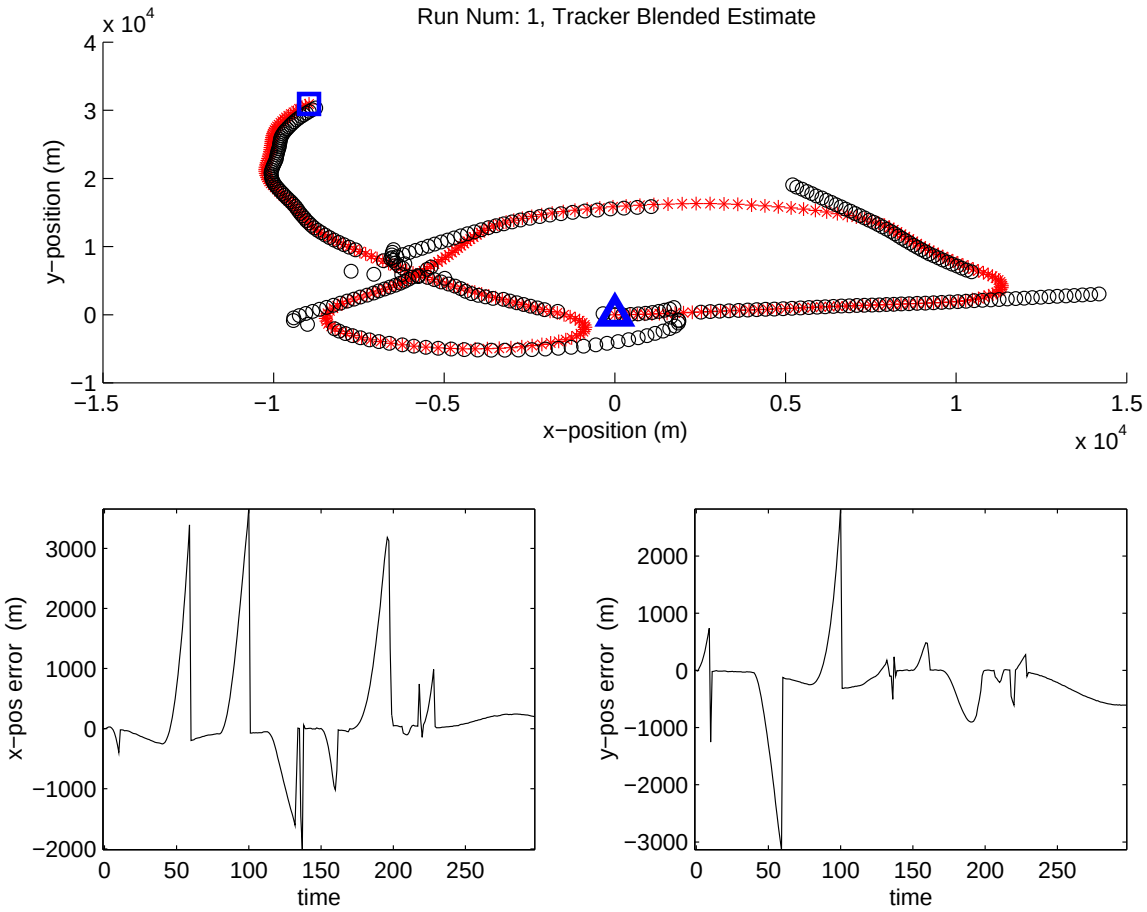

Figure 4.89: Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

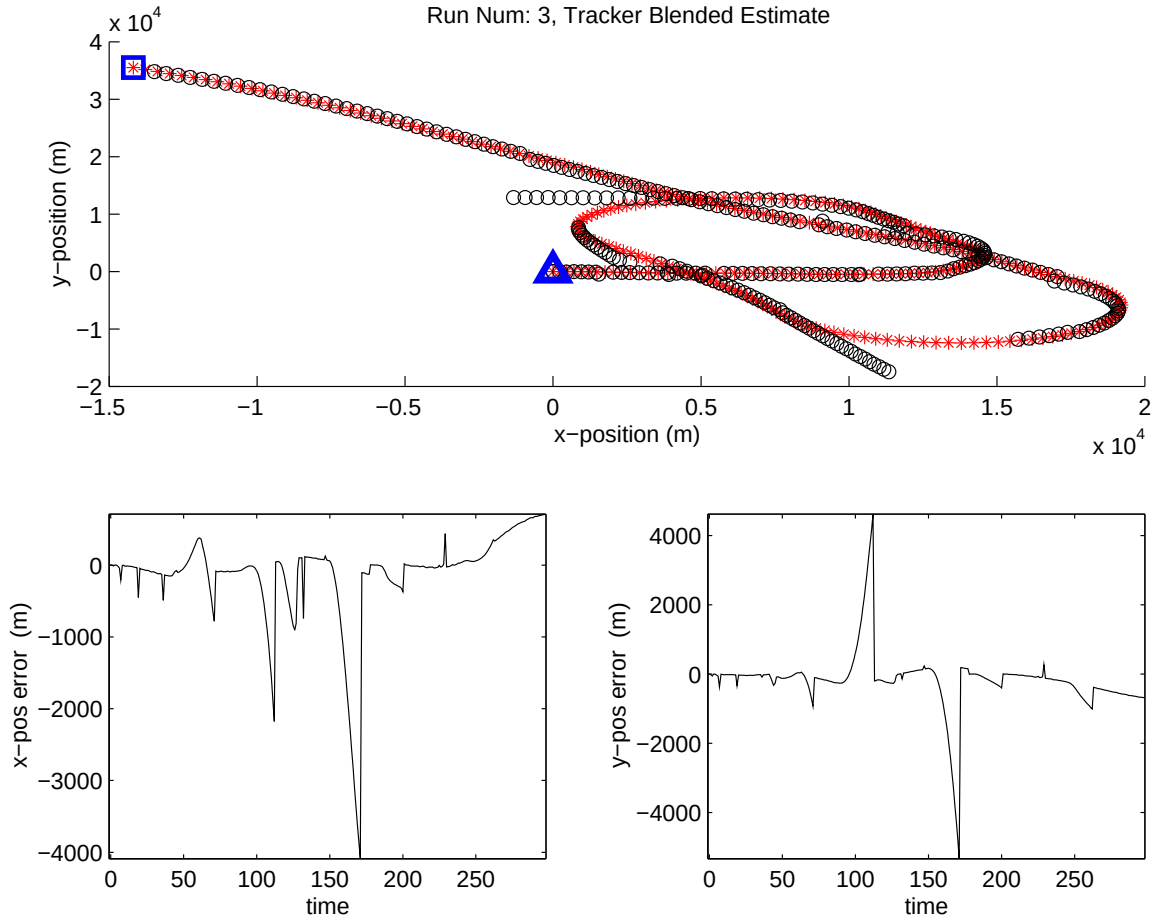

Figure 4.90: Blended estimate for run number 3 in which truth includes three, +3-g TPV maneuvers. Run number 3 is a "best case" run for this MMAE-based suite/scenario combination. (Suite 6 vs. Scenario 6: MMAE)

The summary plots for all 10 Monte Carlo runs (Figs. 4.91 through 4.95) tell a better, and more complete, story. Notice that the blended estimate exhibits good performance and never diverges. The probability weights are not quite as expected, because filter 3, the benign FOGMA filter, picks up *all* of the probability weight from $(k = 80, \ldots, 120)$, which is precisely the portion of flight in which the 40 second, +3-g TPV maneuver is executed. Again, the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ values on filter 1 drop considerably during all of the TPV maneuvers (an excellent indication that filter 1 is a good match to the TPV truth model in force), but the probability flow seems to lag behind these crisp changes by a considerable number of sample periods. Compare this to the cases with Suite 5 running against Scenario 5 (Fig. 4.72) in which the MMAE probability flow is well-synchronized with the onset of the TPV maneuvers. In the Kalman-filter-based case (Fig. 4.35), Suite 6 versus Scenario 6 exhibited better response, although, as discussed in Section 4.1.6, modelling issues play a role in all of the Scenario 6 trials.

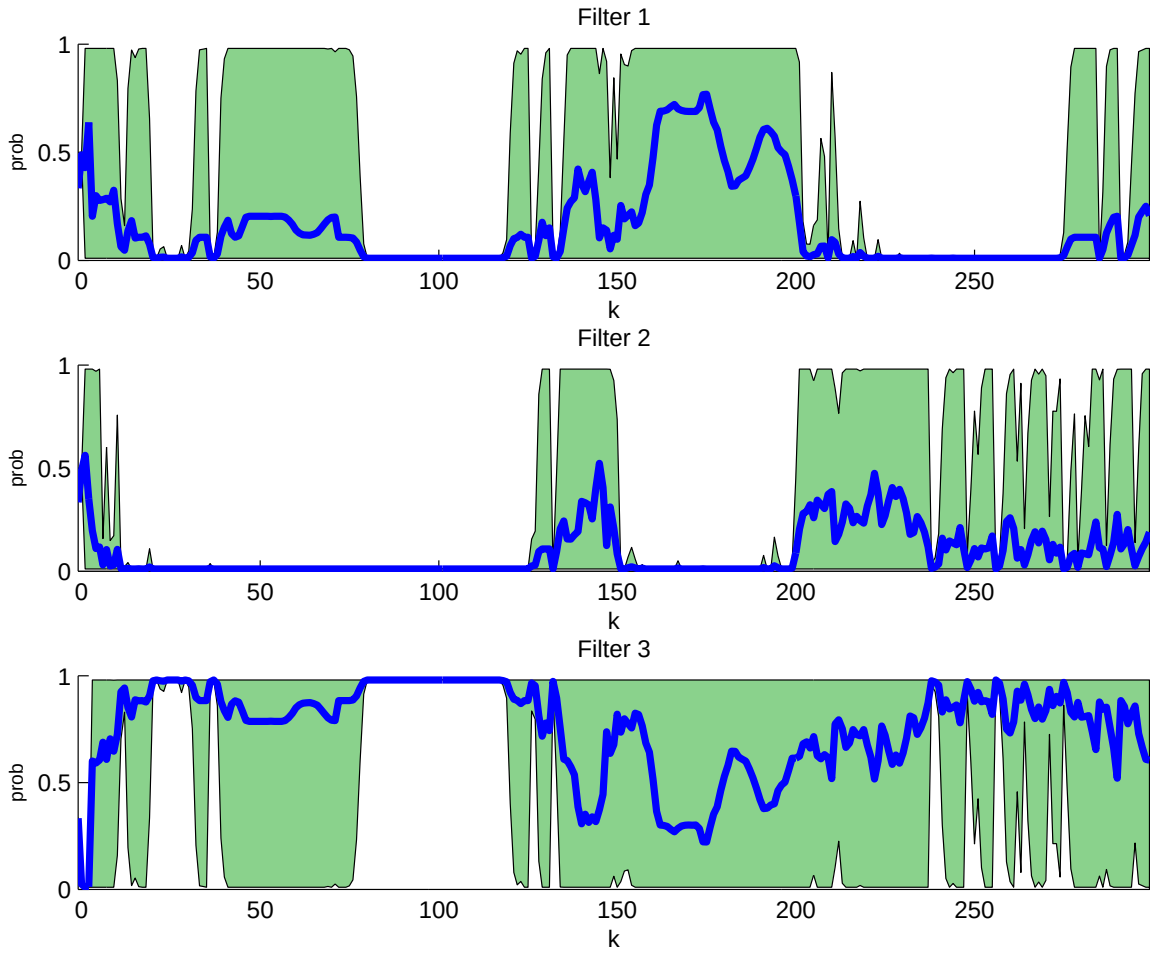

Figure 4.91: Probability flow for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6)

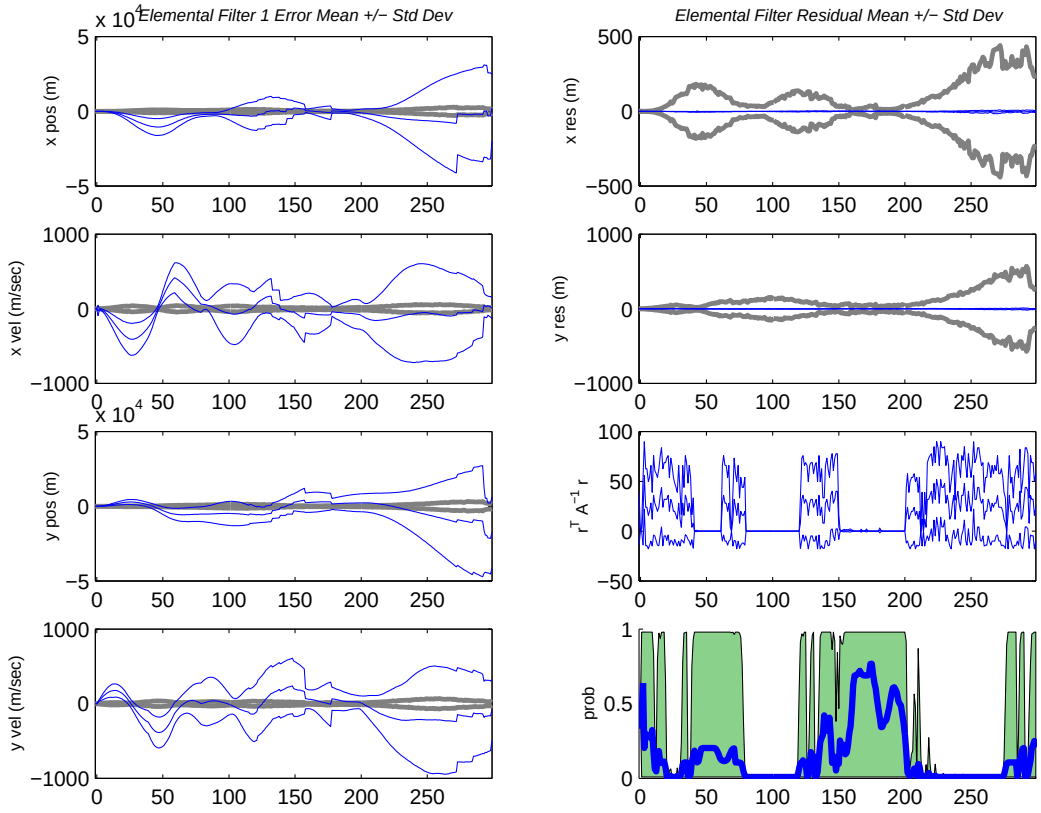

Figure 4.92: Filter 1 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

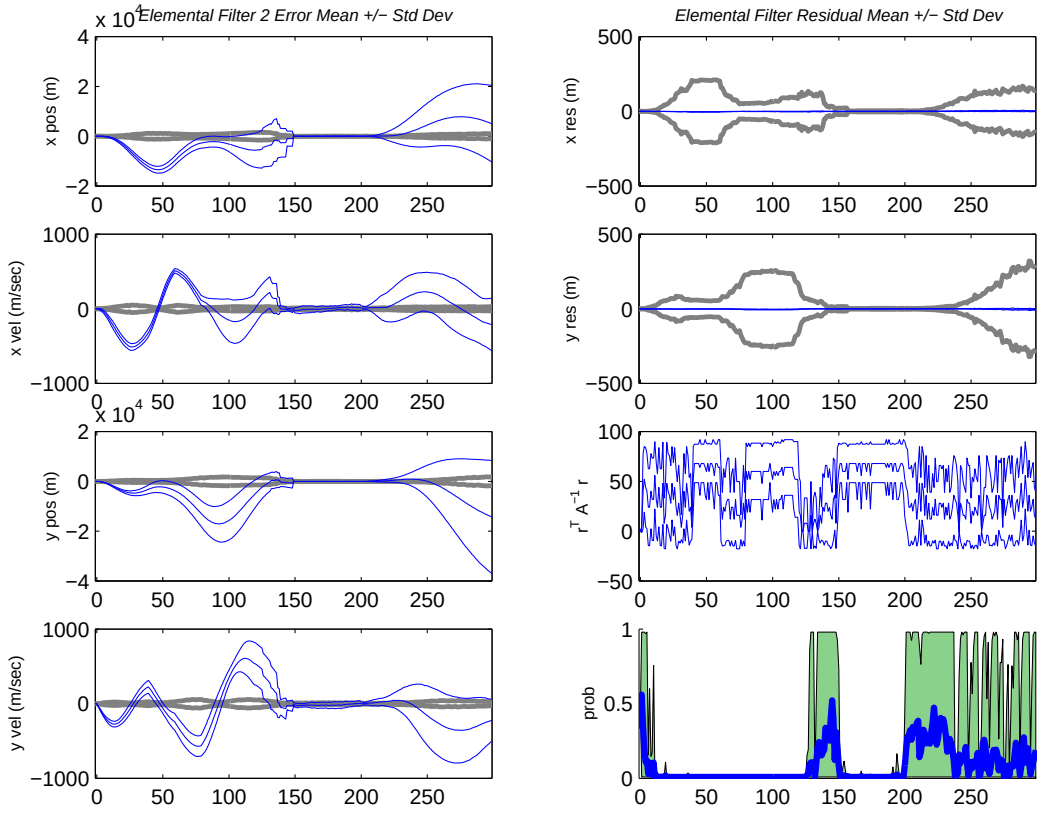

Figure 4.93: Filter 2 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

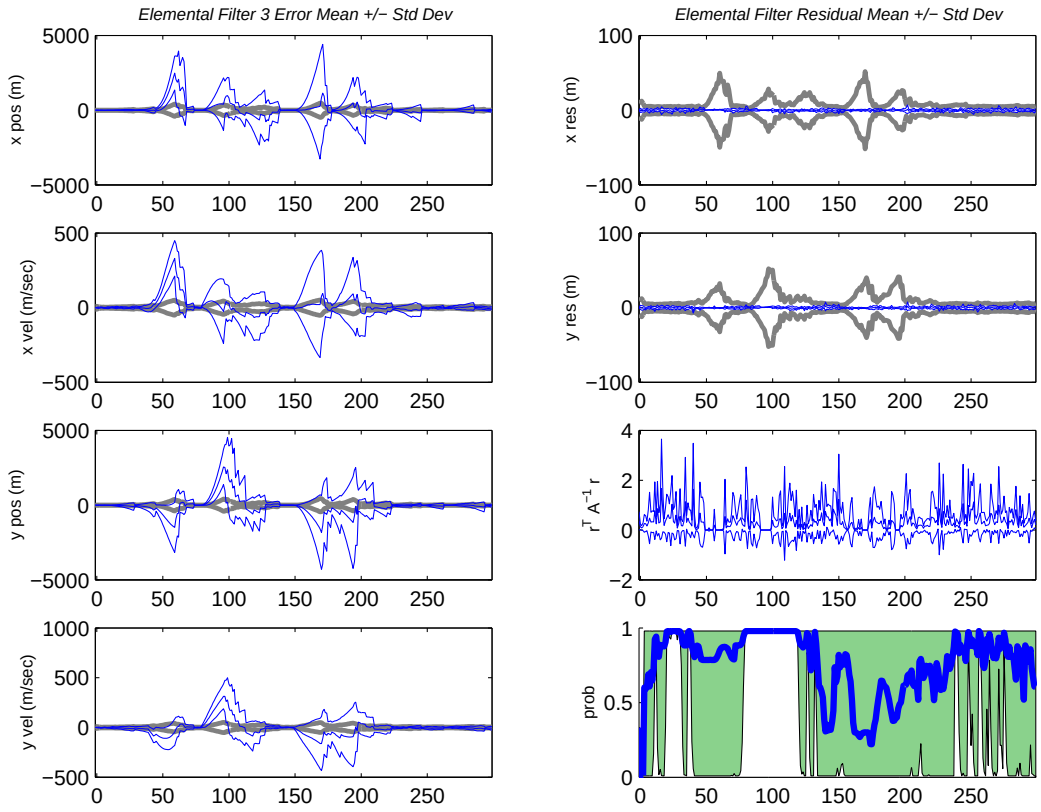

Figure 4.94: Filter 3 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

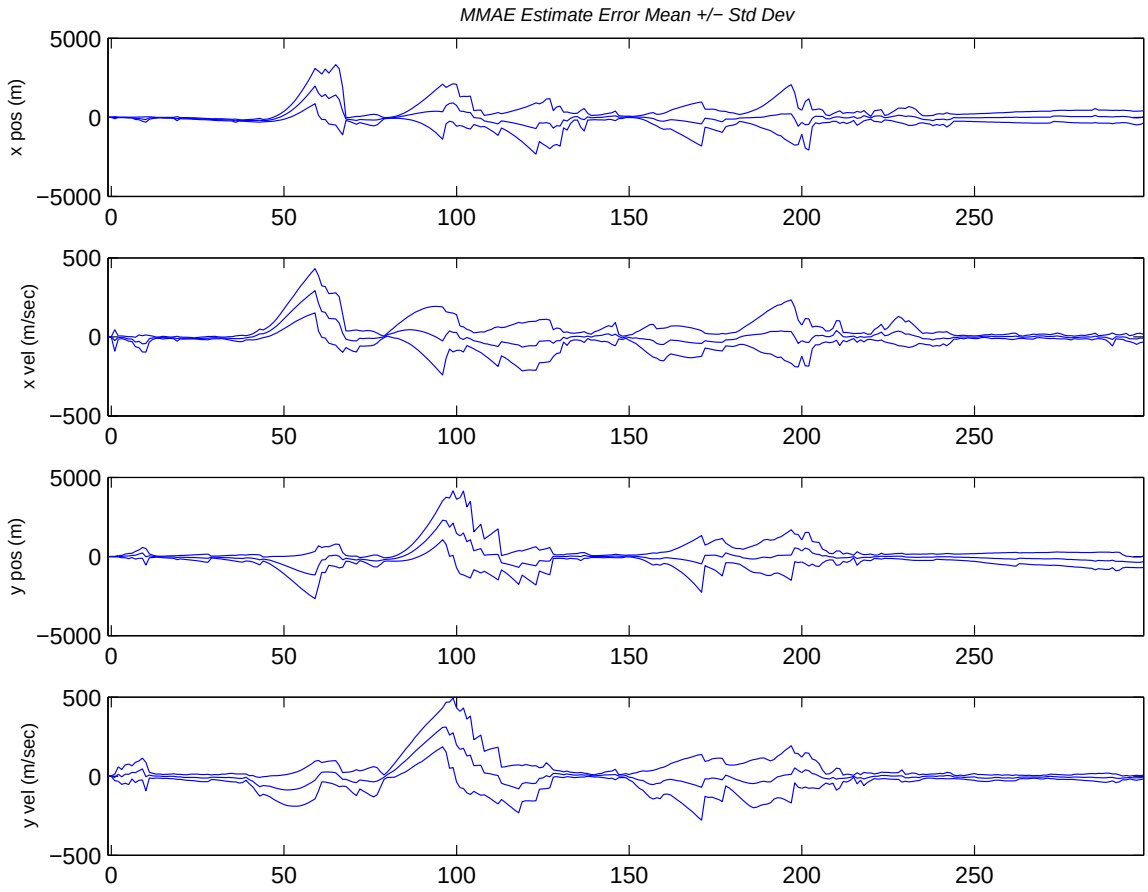

Figure 4.95: Blended estimate data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

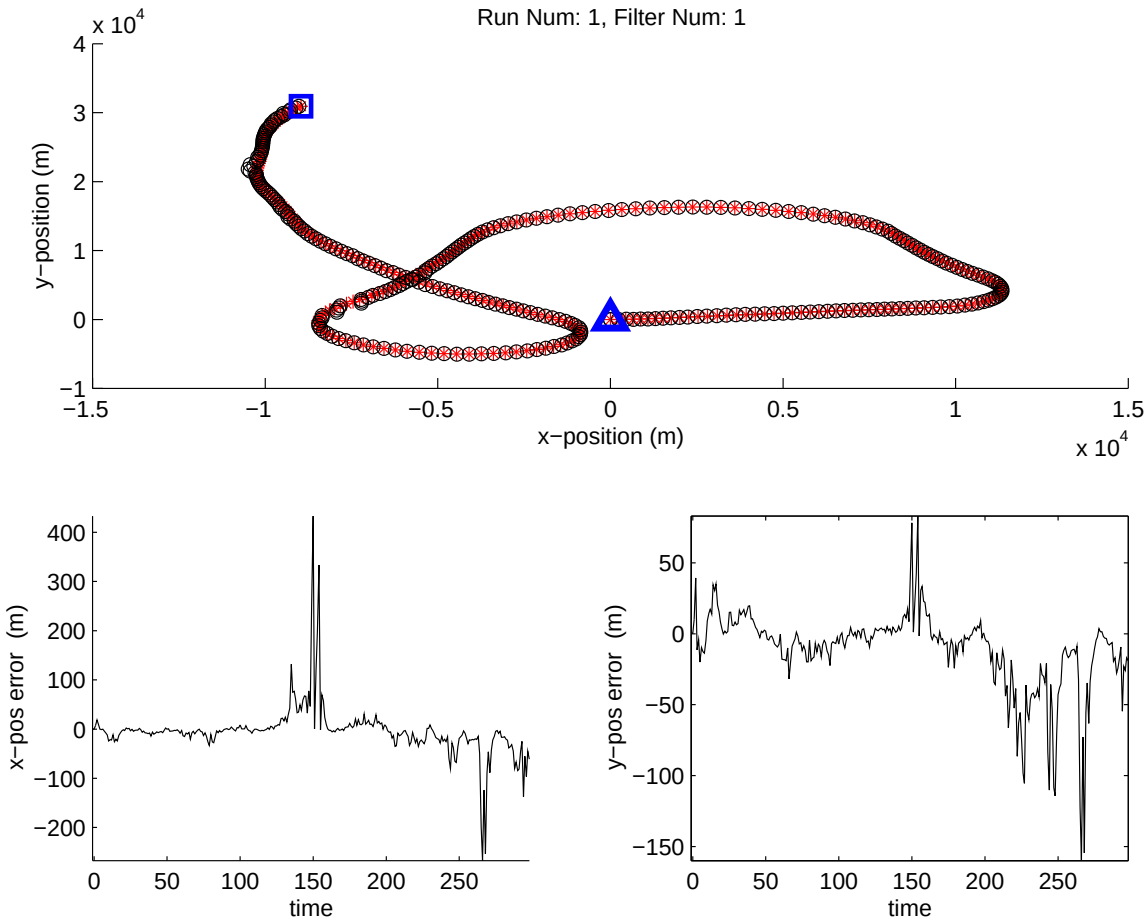

Figure 4.96: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

The IMM blended estimate clearly benefits, in this case, from the continual mixing of state estimates. Figures 4.96 through 4.99 show outputs for the first Monte Carlo run. Notice how all filter estimates remain relatively close to truth trajectory, although keep in mind this is only one of ten Monte Carlo runs. In fact, run 1 represents a best-case, of sorts, for this IMM configuration, whereas it was not the MMAE's best case. As stated above, the MMAE's best case appeared to be run number 3 (Fig. 4.90 shows the blended estimate), whereas the IMM did rather poorly on the same run (Fig. 4.100 shows the blended estimate).

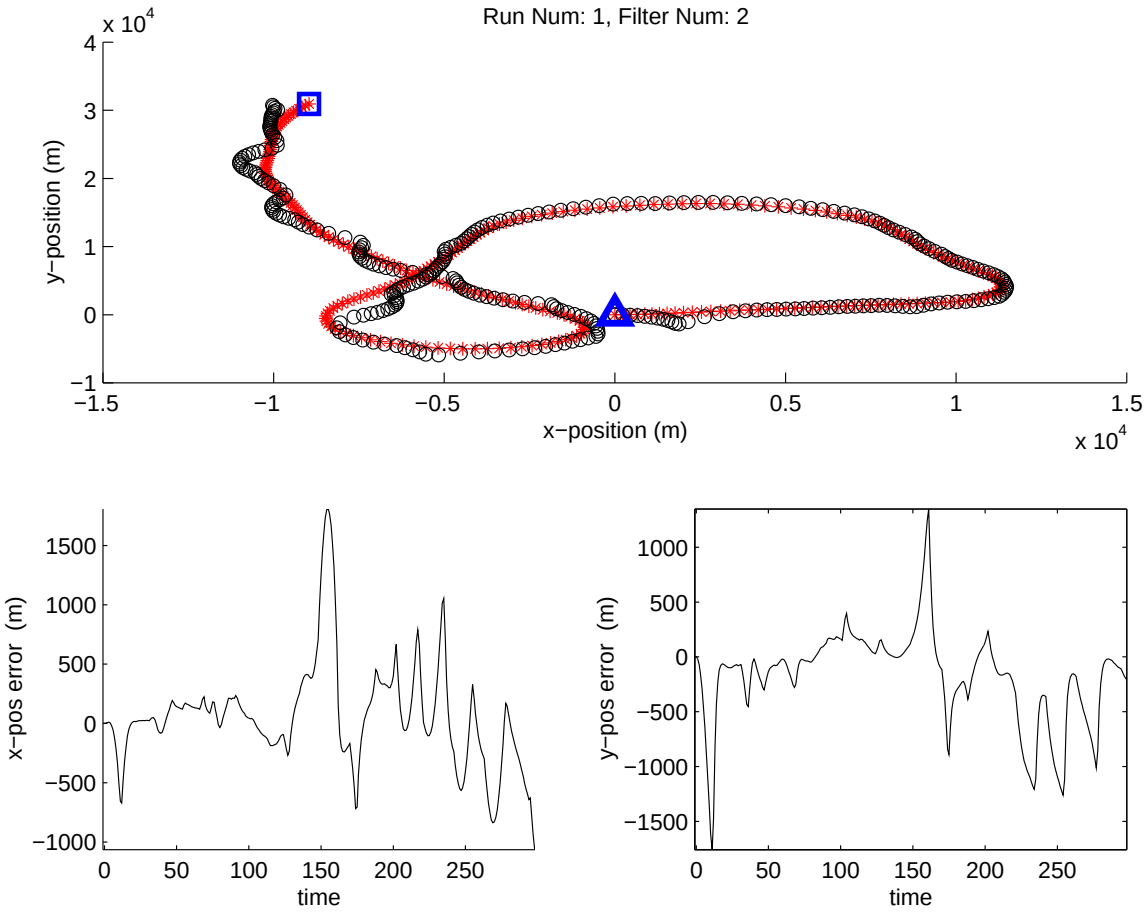

Figure 4.97: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

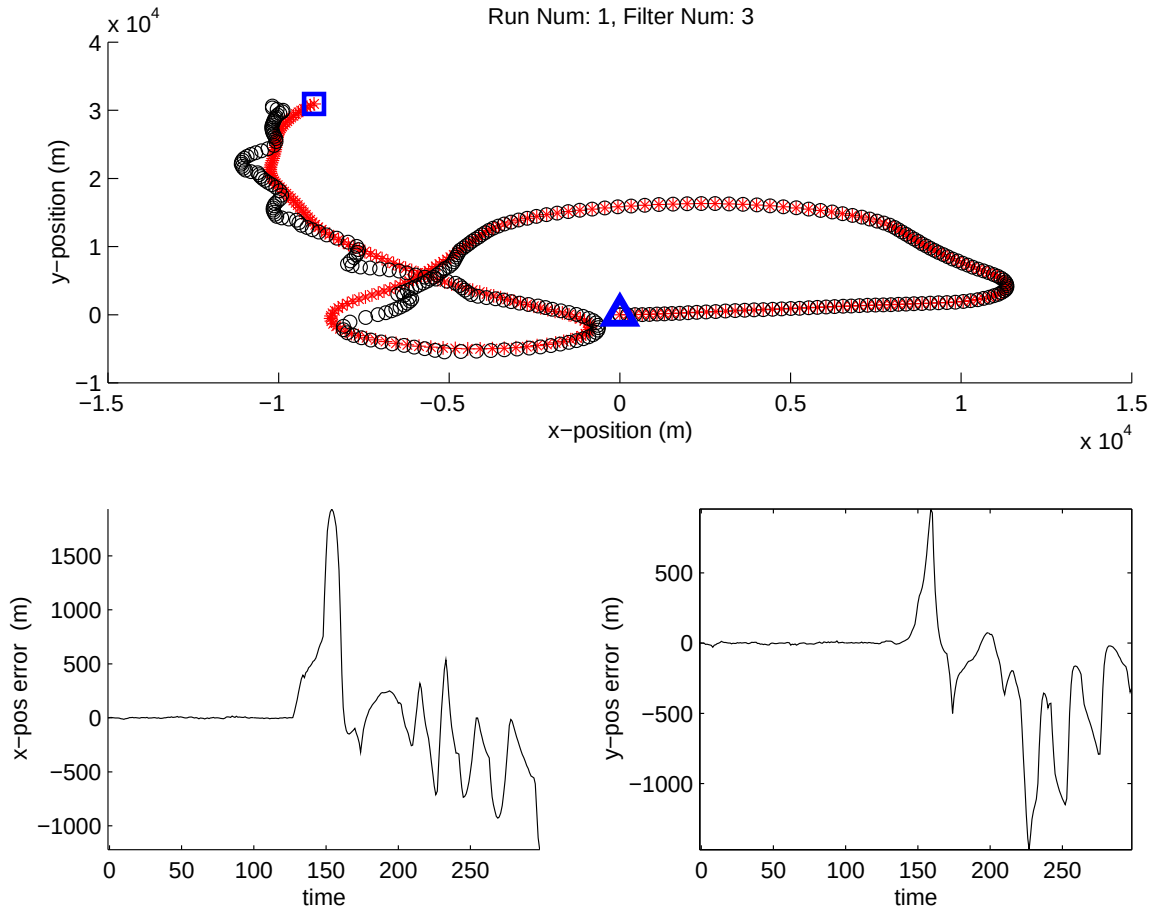

Figure 4.98: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

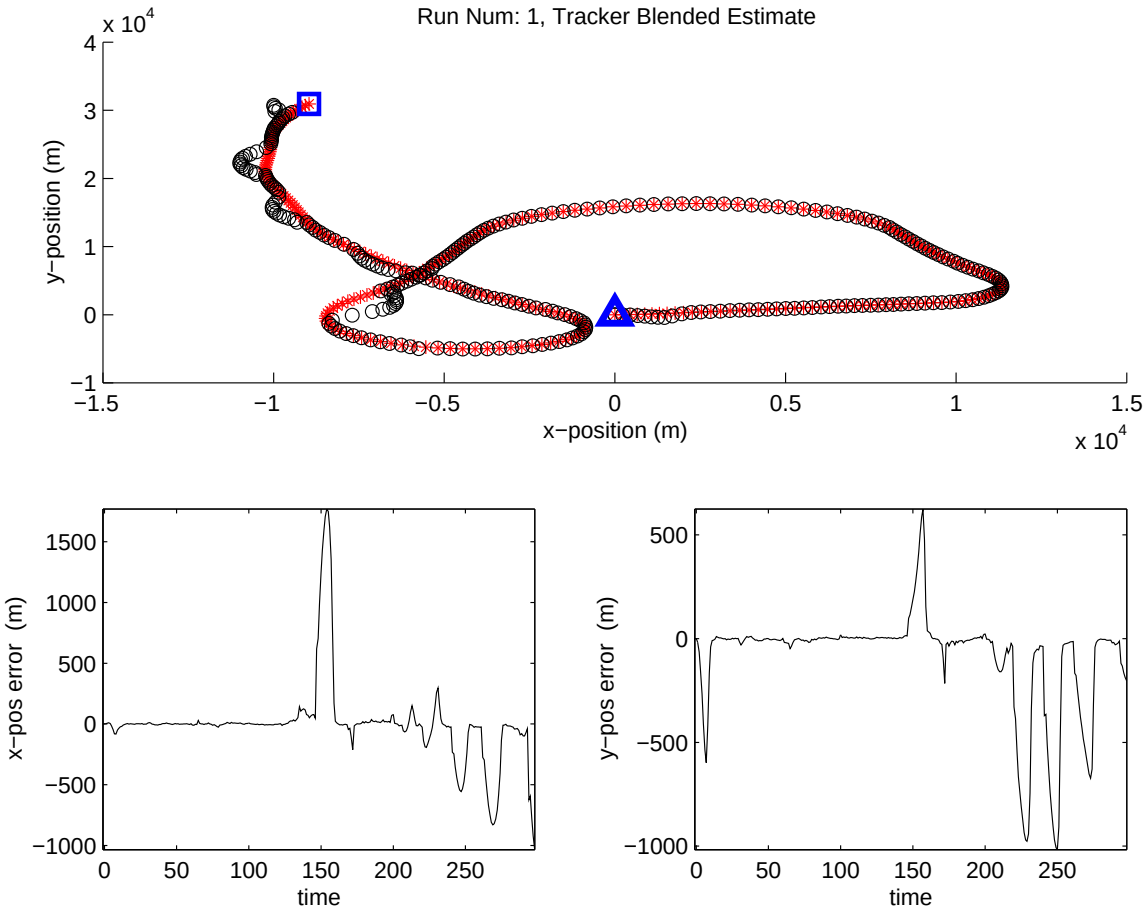

Figure 4.99: IMM Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

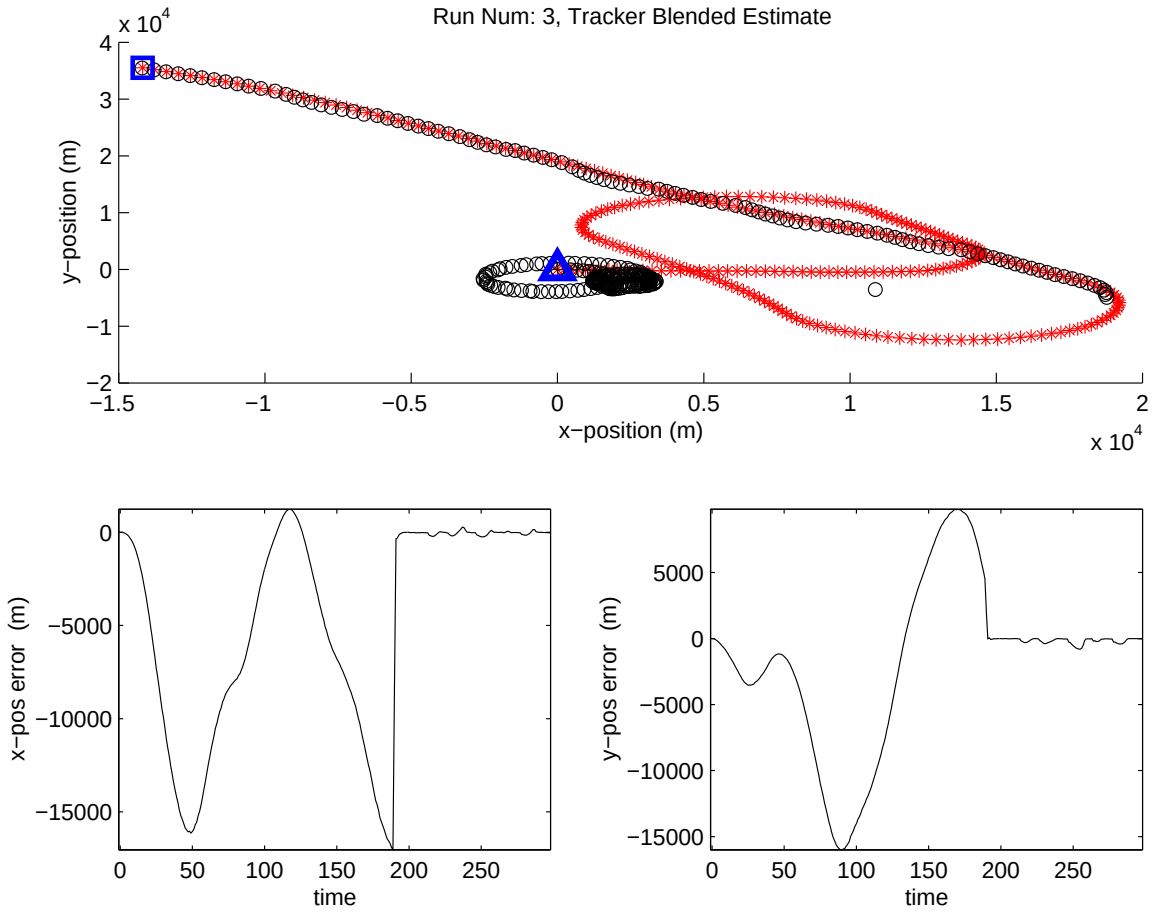

Figure 4.100: IMM Blended estimate for run number 3 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

Interestingly, the IMM performed considerably less well when evaluated over the set of 10 Monte Carlo runs. Figures 4.101 through 4.105 show summary plots for this case. Notice the probability flows are characteristically ambiguous in the IMM (Fig. 4.101), while MMAE's appear to correlate more with the truth maneuvers (see Fig. 4.91). Also, the MHT IMM's probability flows differ considerably from the no-clutter, Kalman-filter-based case (shown in Fig. 4.37).

Compared to the MMAE's blended estimate (shown in Fig. 4.95), the IMM's blended estimate mean $\pm 1\sigma$ error plots (Fig. 4.105) have a much larger envelope, implying the blended solution error varied considerably more across the 10 runs than it did for the MMAE. A tighter envelope, as shown by the MMAE, would suggest the tracker's state estimate error performance is more repeatable over the 10 trials. It would also appear that the IMM tracked the target less well during the non-maneuvering phases, evidenced by the higher magnitude $\pm 1\sigma$ excursions (notice the difference in the plotting scales for the MMAE and IMM). The smaller magnitude excursions in the MMAE results would suggest that, although the error is not always zero mean (the mean error taken over all 10 runs at the same sample period), the absolute error committed is generally smaller than that produced by the identically configured IMM.

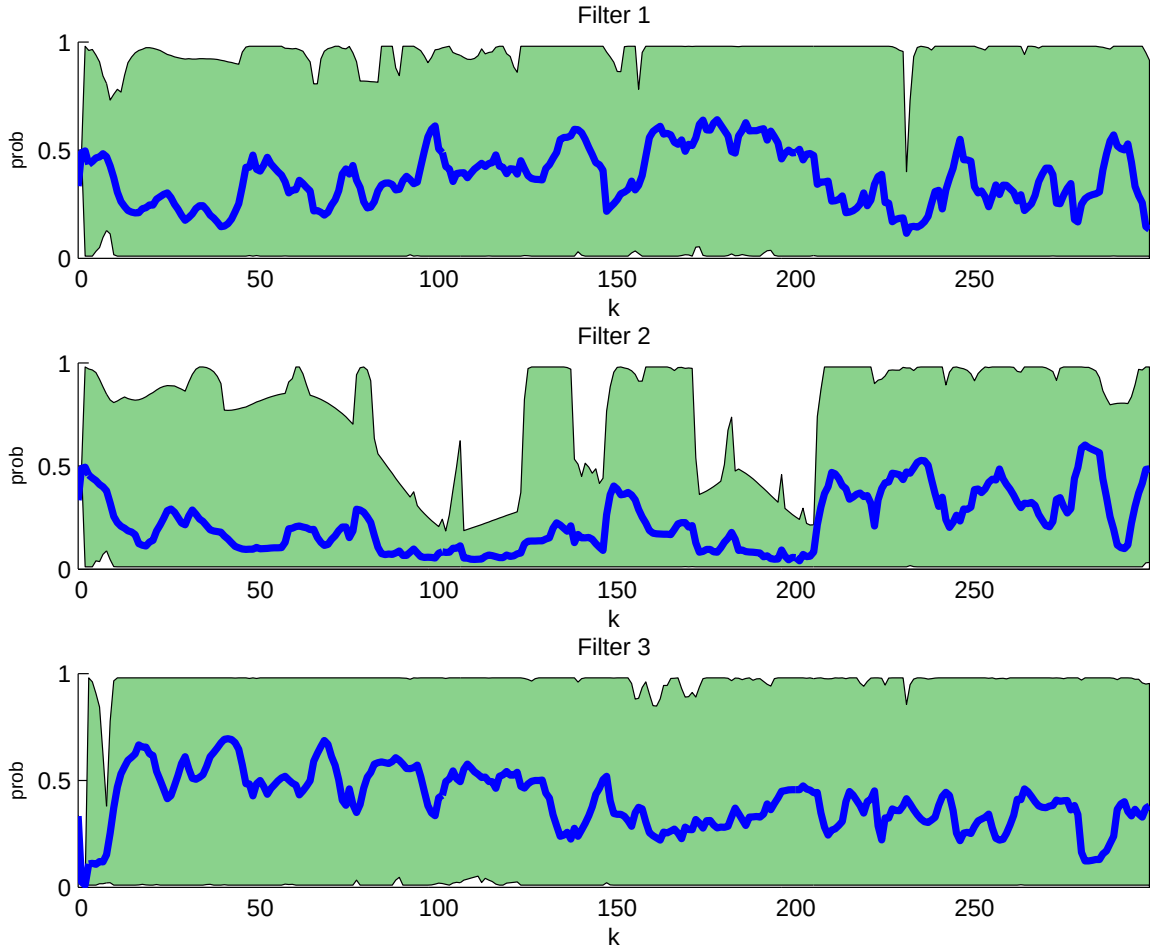

Figure 4.101: Probability flow for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

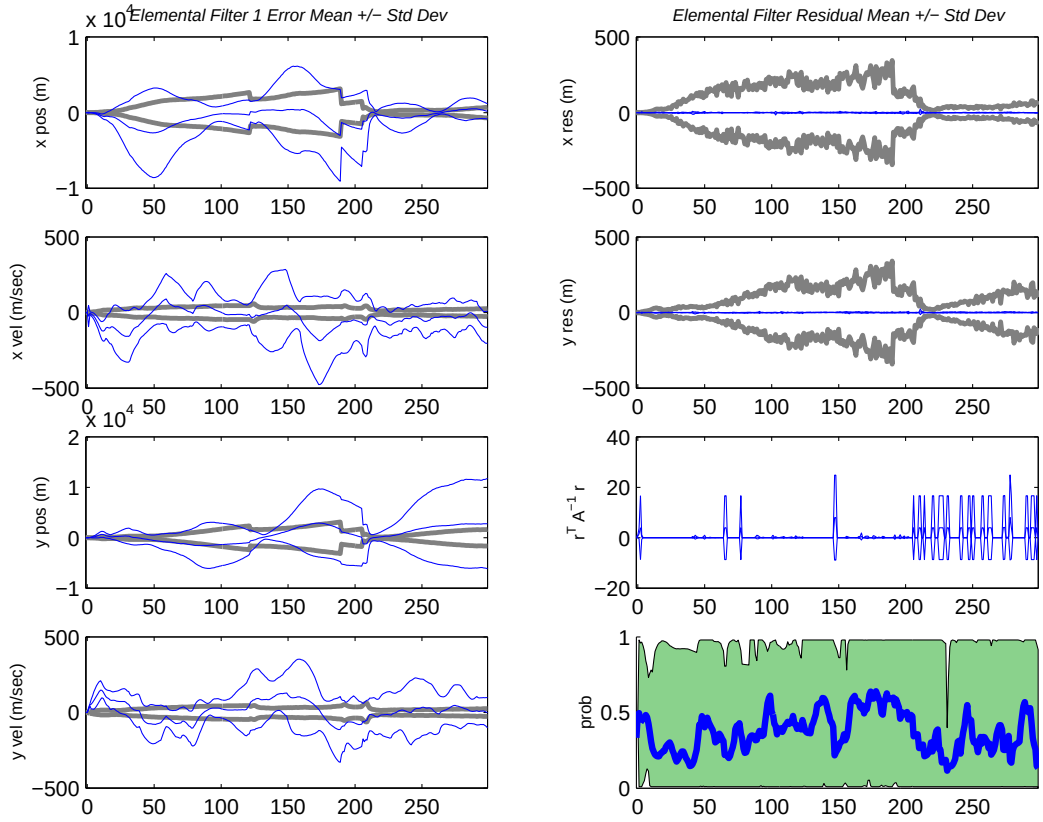

Figure 4.102: Filter 1 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

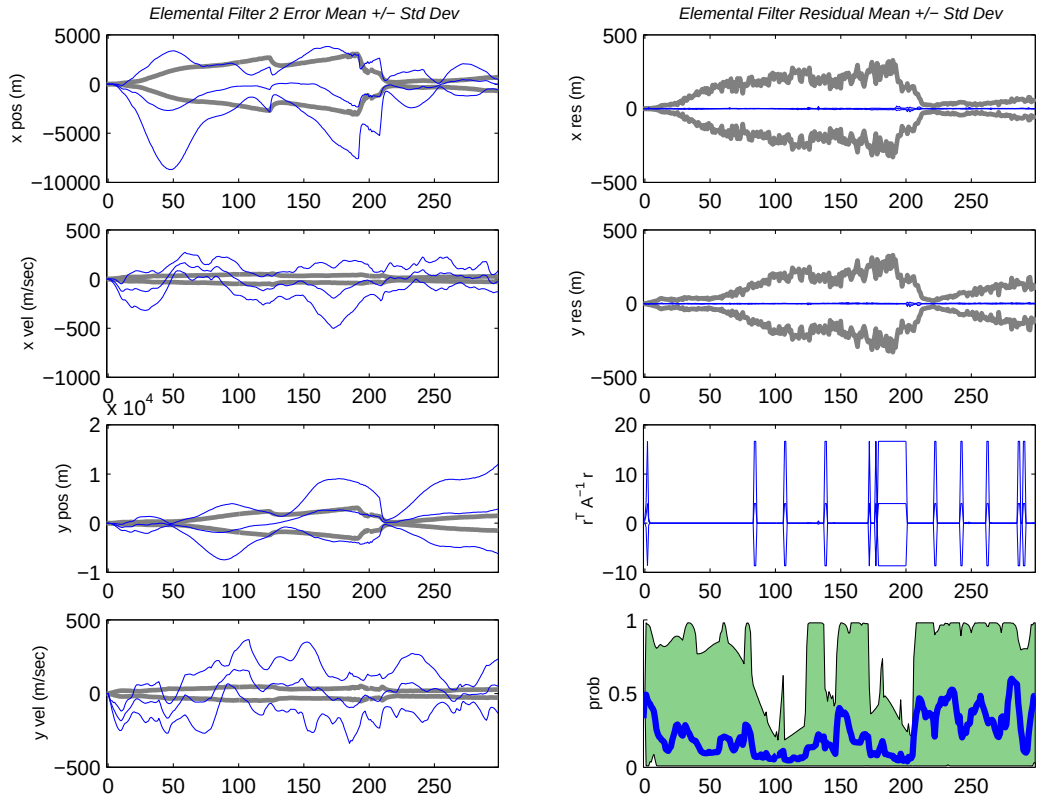

Figure 4.103: Filter 2 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

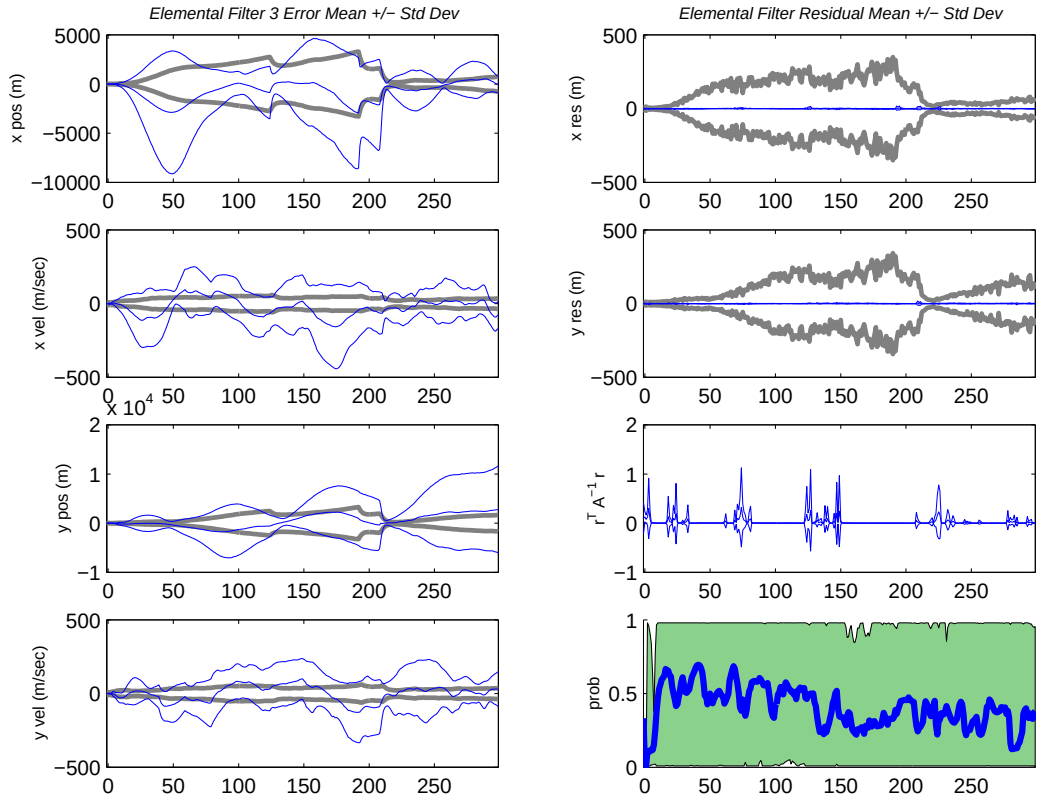

Figure 4.104: Filter 3 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

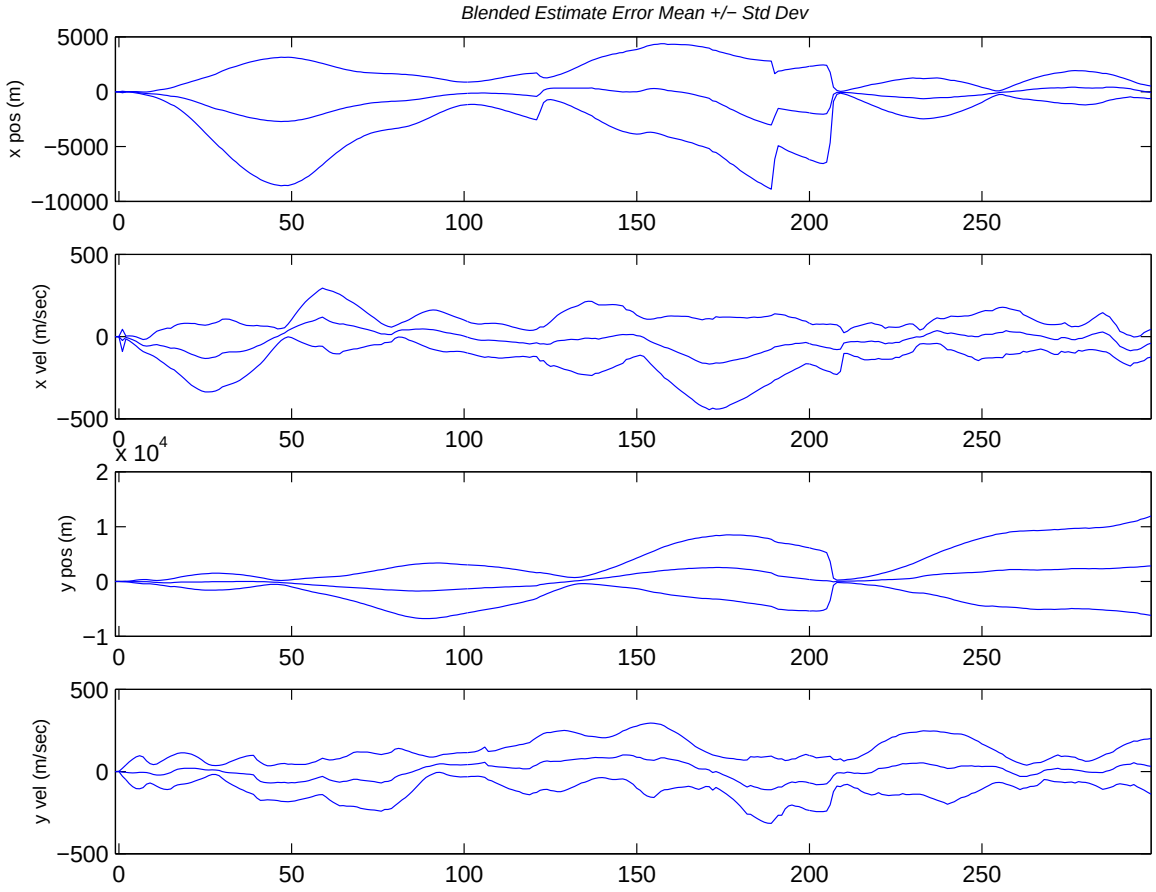

Figure 4.105: IMM Blended estimate data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

4.5 Medium Clutter Density Results

The highest measurement clutter density used in this research yielded an expected 10 measurements per cycle with a probability of detection equal to 1.0 (ensuring the true measurement would exist in every measurement scan). The maximum number of components existing after the Williams mixture reduction cycle was specified to be 20. In theory, as clutter density increases, there is considerable merit in allowing a larger number of mixture components to exist after mixture reduction, because the additional measurement association hypothesis histories would aid the MHT's deferred decision-making. However, initial tests with the limit set to 40, and then 30, components proved to be too time consuming and extremely resource intensive.

Whereas the low-clutter case with 5 measurements per cycle had the mixture limit set to 30, the increased computational demands, as well as the increased memory storage demands, of the 10 measurements per cycle case forced the limit to be reduced. By allowing fewer mixture components to carry across algorithm cycles, the number of components being matched to the increased number of measurements was effectively balanced. For example, in the low-clutter case (assuming a system with only one elemental filter), the number of components existing after measurement association would be:

$$\begin{aligned} N_h(k) &= N_r(k-1) \cdot N_z(k) + N_r(k-1) \\ &= 30 \cdot 5 + 30 = 180 \end{aligned}$$

where $N_h(k)$ is the number of components existing after measurement association but prior to mixture reduction, $N_r(k-1)$ is the number of components carried over from the previous algorithm cycle, $N_z(k)$ is the number of incoming measurements, and the additional $N_r(k-1)$ term reflects the missed detection hypotheses (which, by definition, do not receive a measurement update). Conversely, using the mixture limit of 20 at 10 measurements per cycle:

$$\begin{aligned} N_h(k) &= N_r(k-1) \cdot N_z(k) + N_r(k-1) \\ &= 20 \cdot 10 + 20 = 220 \end{aligned}$$

Therefore, the medium-clutter case is designed to show that the multiple-model MHT algorithms using Williams elemental filters can effectively handle a case with more clutter measurements (i.e., cases with a larger number of *false* measurement association histories). A larger mixture limit than that used in the low-clutter case *would* help the MHT's deferred decision-making ability, but computer limitations (speed of execution and memory storage) necessitated a *reduction* in that limit. From an algorithm assessment standpoint, though, the cases shown here represent a greater chal-

lenge than the case of increasing the measurement clutter density while leaving the mixture limit unchanged or making it larger. In other words, the importance placed on the Williams mixture reduction algorithm performance will be *greater*, because the MHT will not have the benefit of the increased number of hypotheses allowed by the higher mixture component limit.

Finally, the reader should note two differences in the analysis of the medium-clutter simulation trials. First, long simulation run times allowed only a limited number of filter/truth configurations to be tested. Many of those trials were not complete enough to be of analytical value, so only three configurations are shown here. Suites 4 and 6 include both MMAE and IMM results, whereas Suite 5 includes only MMAE. Secondly, changes to the simulation software after the completion of the Kalman filter, ultra-low-clutter, and low-clutter analyses caused Monte Carlo seed numbers to be handled slightly differently than they were in the previous trials. Therefore, meaningful direct comparison of single Monte Carlo run plots between the medium-clutter results and previous sections is not possible. However, the analysis concepts used to interpret these single-run plots will be identical, and the 10 run summary plots should be comparable across all clutter cases.

The reader is reminded that the track loss checks discussed in Chapter 3, Section 3.4 will be treated in Appendix C. The emphasis here will be on qualitative analysis of tracker performance, whereas the appendix emphasizes a more quantitative approach to the analysis.

4.5.1 Suite 4 vs. Scenario 4. These simulations pitted a filter bank containing two FOGMA filters against a piecewise time-varying FOGMA truth model. Both MMAE and IMM performed better than expected, and overall performance was comparable to the cases with only 5 expected measurements per cycle. Figures 4.106 through 4.108 show single-run outputs for the MMAE on Monte Carlo trial number 2. Notice the MHT's characteristic snapping of the filter estimate in the MMAE (Fig. 4.106) as it undergoes a reset similar to the situation described for this Suite/Scenario combination in the low-clutter case (see Section 4.4.3).

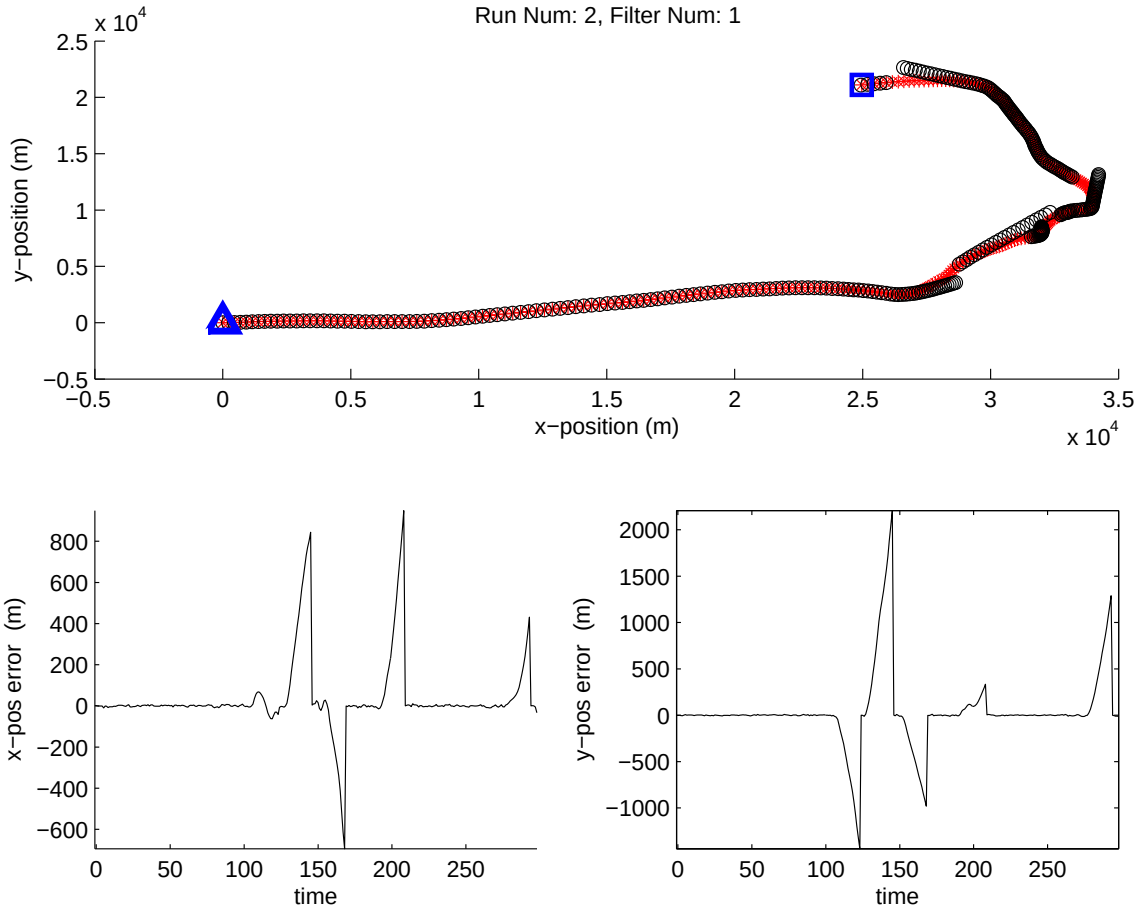

Figure 4.106: Filter 1 solution on Monte Carlo run number 2. (Suite 4 vs. Scenario 4: MMAE)

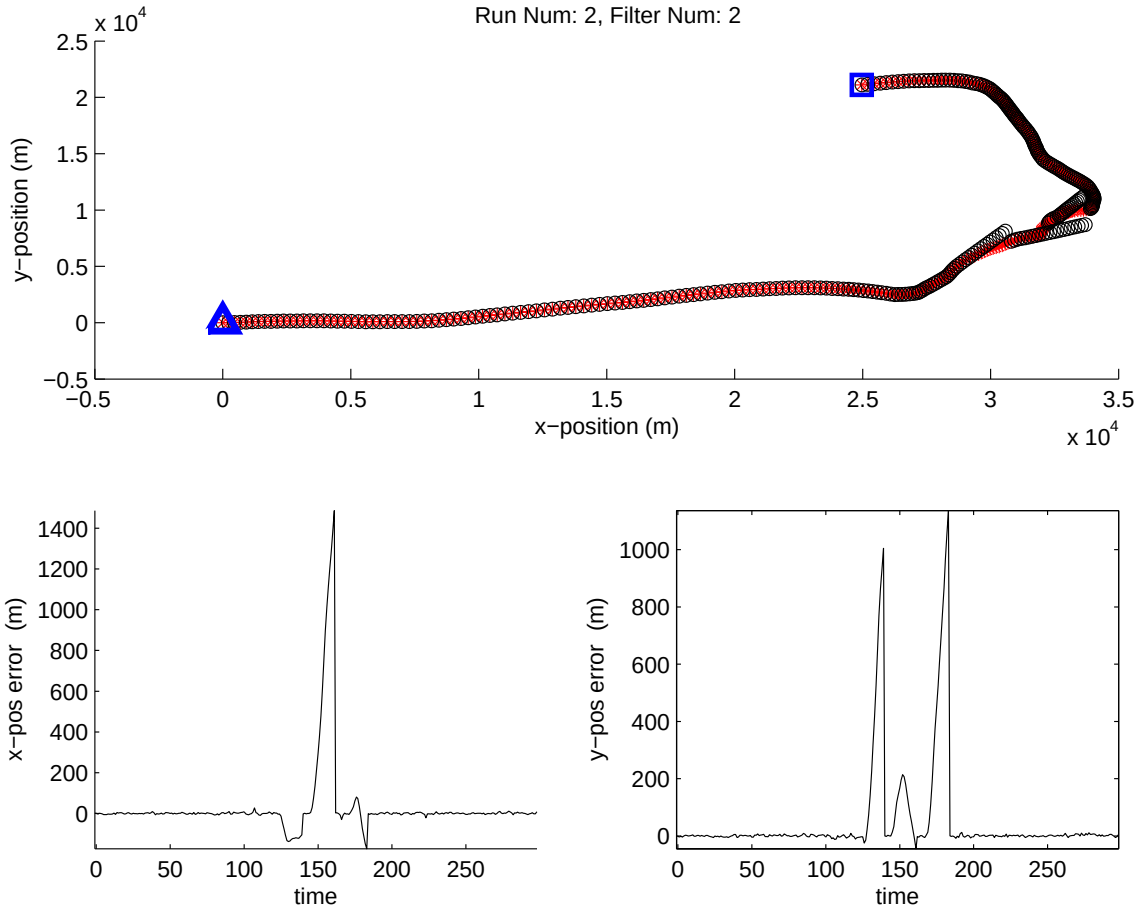

Figure 4.107: Filter 2 solution on Monte Carlo run number 2. (Suite 4 vs. Scenario 4: MMAE)

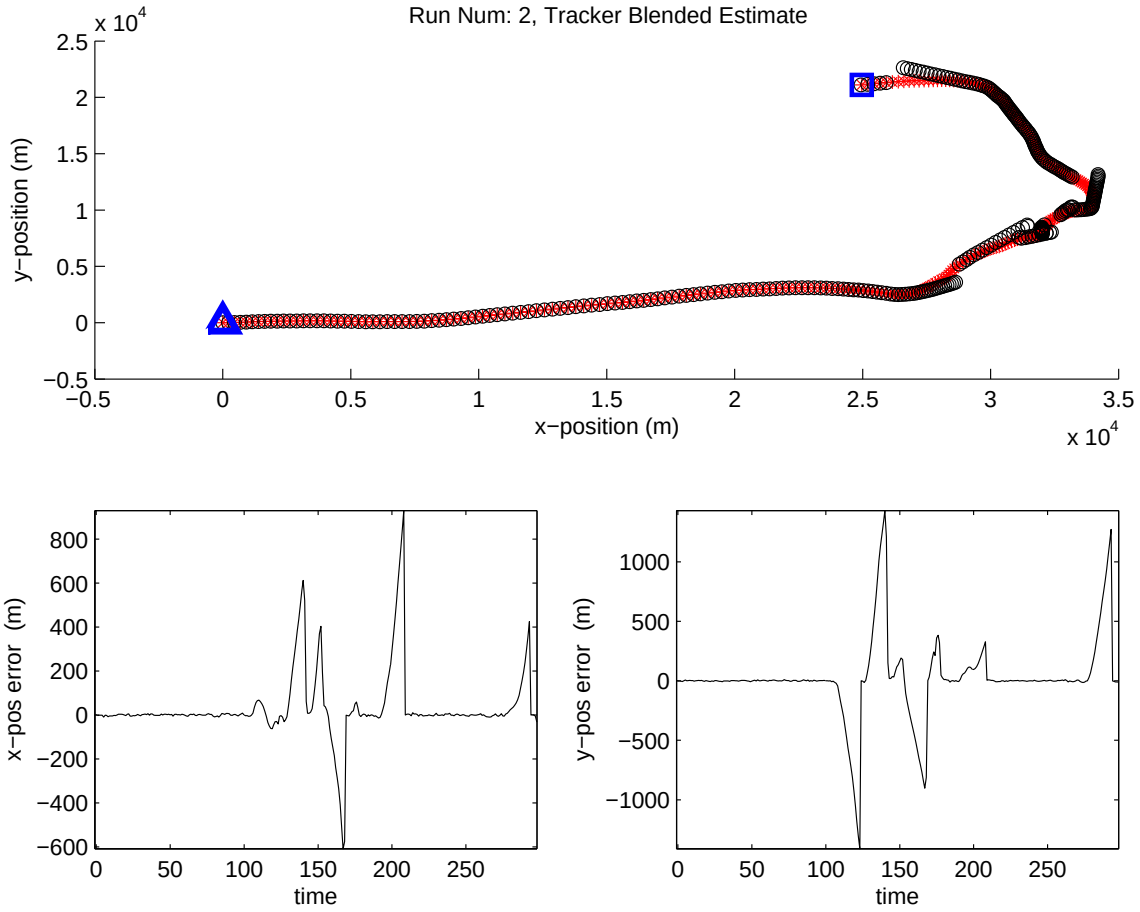

Figure 4.108: Blended solution on Monte Carlo run number 2. (Suite 4 vs. Scenario 4: MMAE)

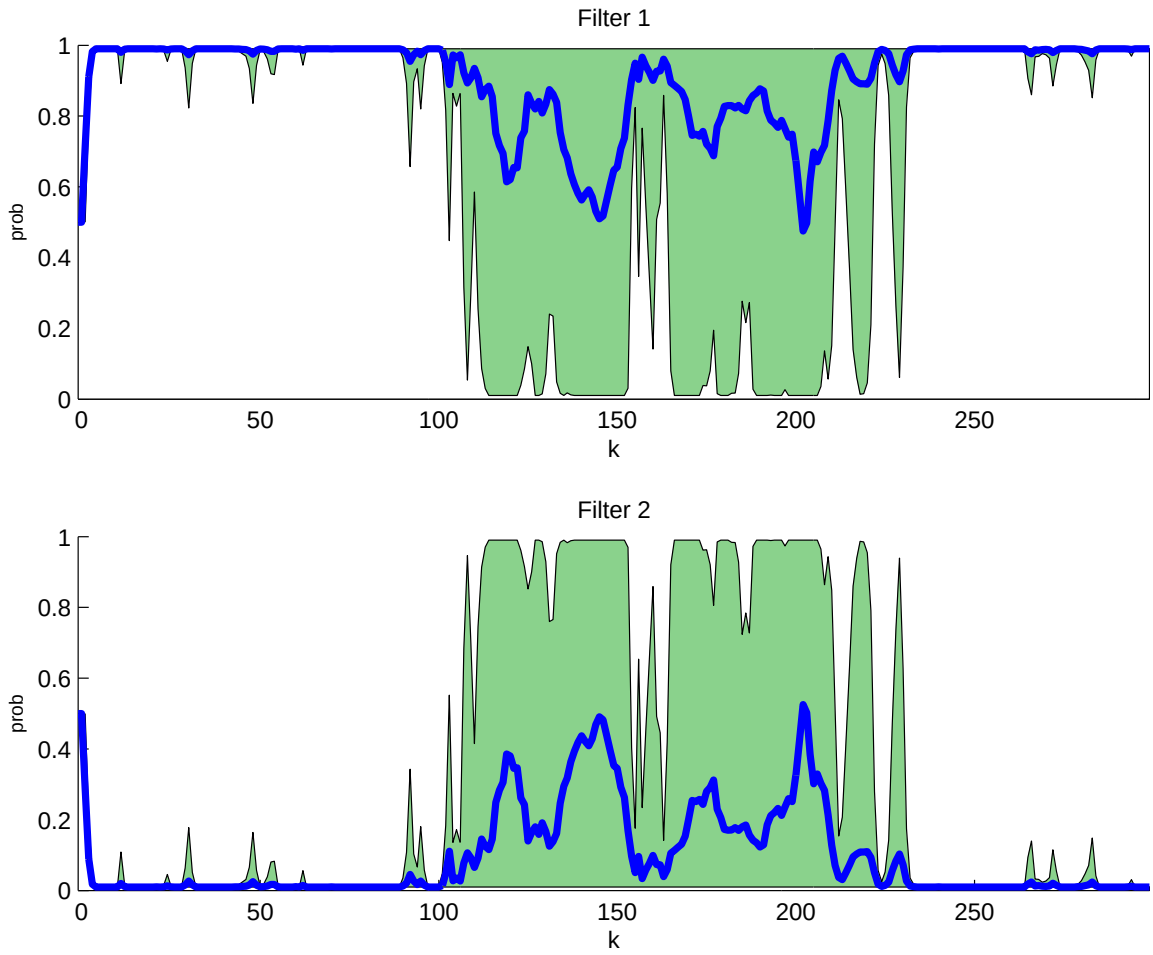

Figure 4.109: Probability flows for the medium clutter density case. (Suite 4 vs. Scenario 4: MMAE)

The summary plots are shown in Figs. 4.109 through 4.112. The probability flows are comparable to those seen in the low-clutter case (Fig. 4.59) but again not as well-defined as those from the no-clutter case (Figs. 4.5 and 4.6). The blended estimate looks very good, and in fact, commits errors on par or better than that seen in the low-clutter case (Fig. 4.58). Note that the MMAE simulation data reflects the full set of 10 Monte Carlo trials, as will the IMM simulation results.

Figures 4.113 through 4.115 show the single-run plots for the equivalent IMM with a non-transition favoring Markov matrix. Like the low-clutter cases, the IMM elemental filters do not exhibit the same discontinuities as the MMAE elemental filters. On this trial, the actual error committed by the filters and the blended estimate are quite small compared to those of the MMAE.

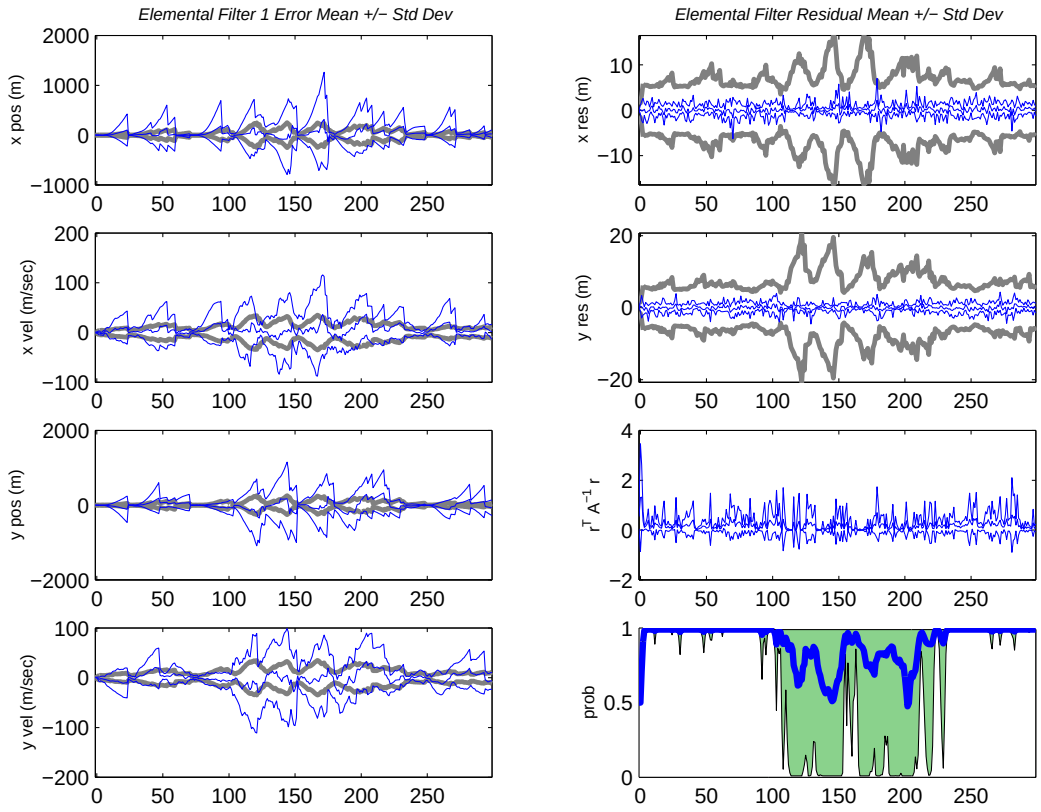

Figure 4.110: Filter 1 data for 10 Monte Carlo runs. (Suite 4 vs. Scenario 4: MMAE)

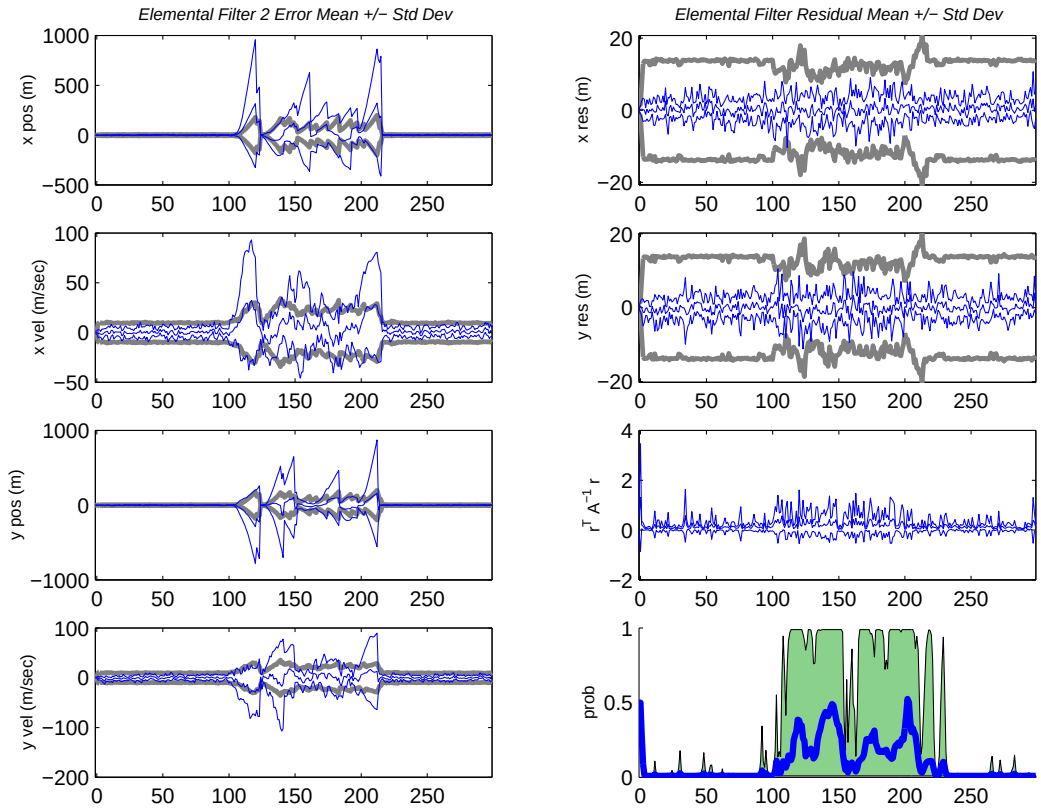

Figure 4.111: Filter 2 data for 10 Monte Carlo runs. (Suite 4 vs. Scenario 4: MMAE)

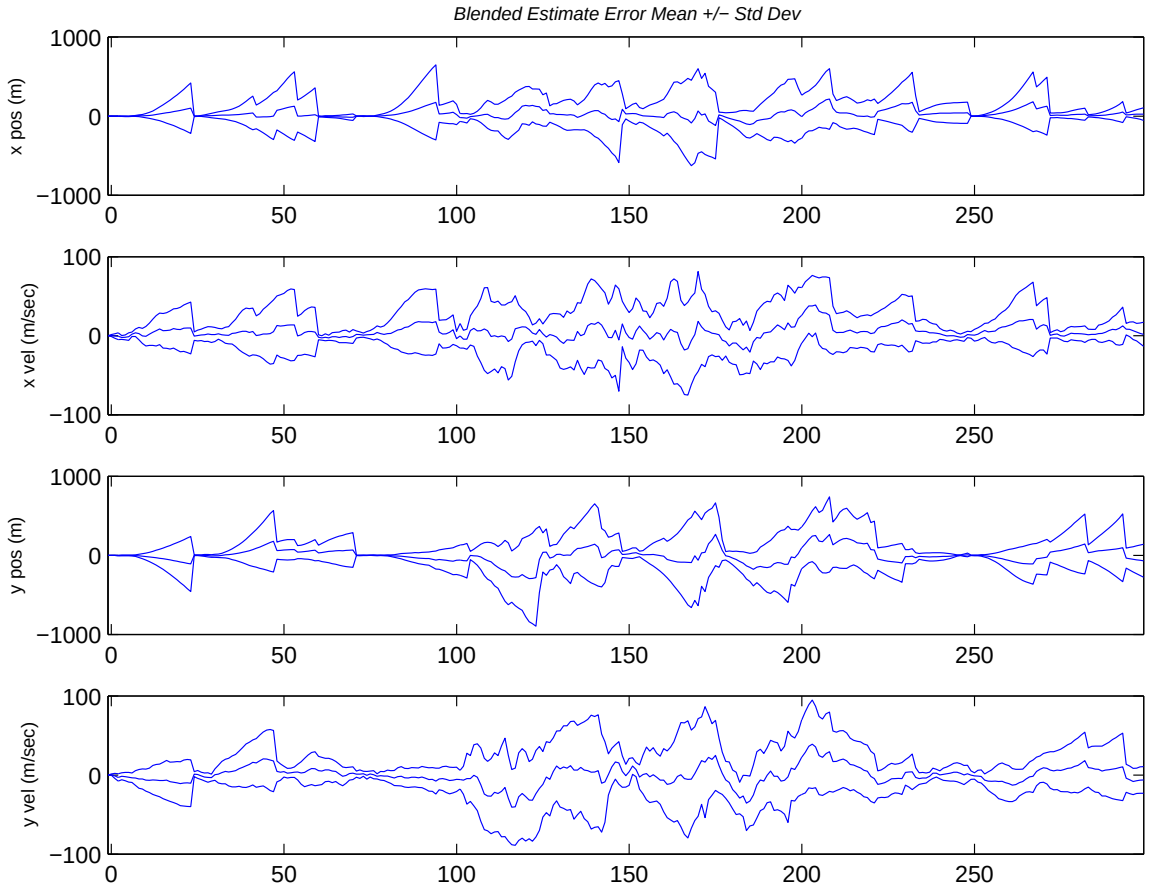

Figure 4.112: Blended estimate for the medium clutter density case. (Suite 4 vs. Scenario 4: MMAE)

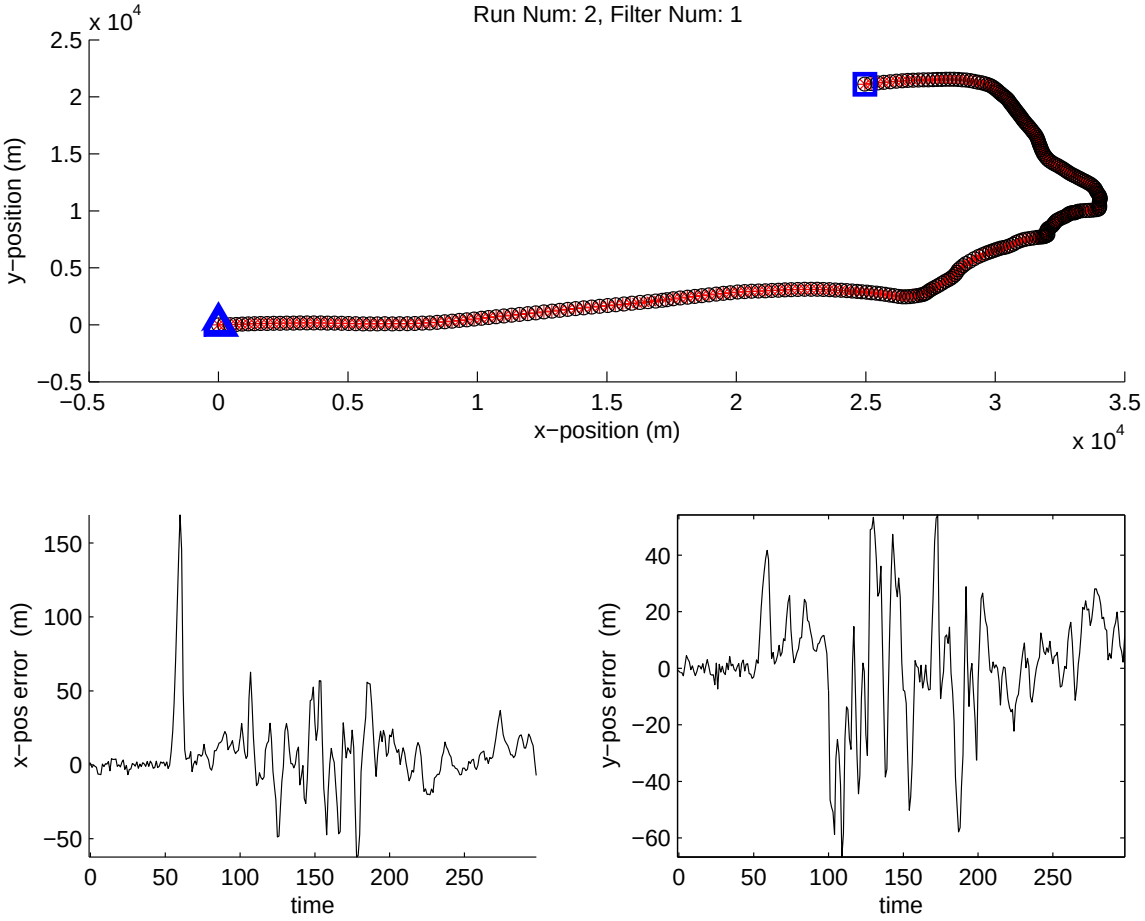

Figure 4.113: Filter 1 solution on Monte Carlo run number 2. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

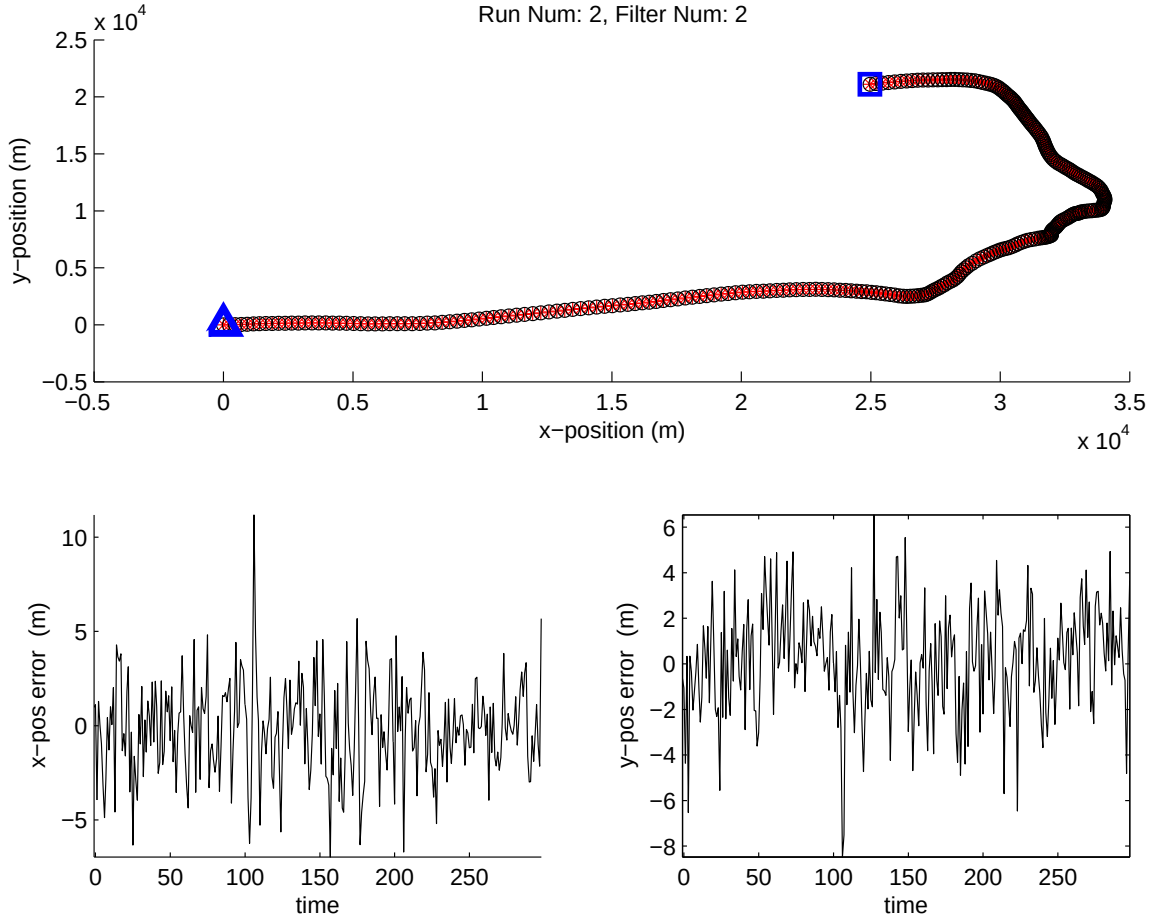

Figure 4.114: Filter 2 solution on Monte Carlo run number 2. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

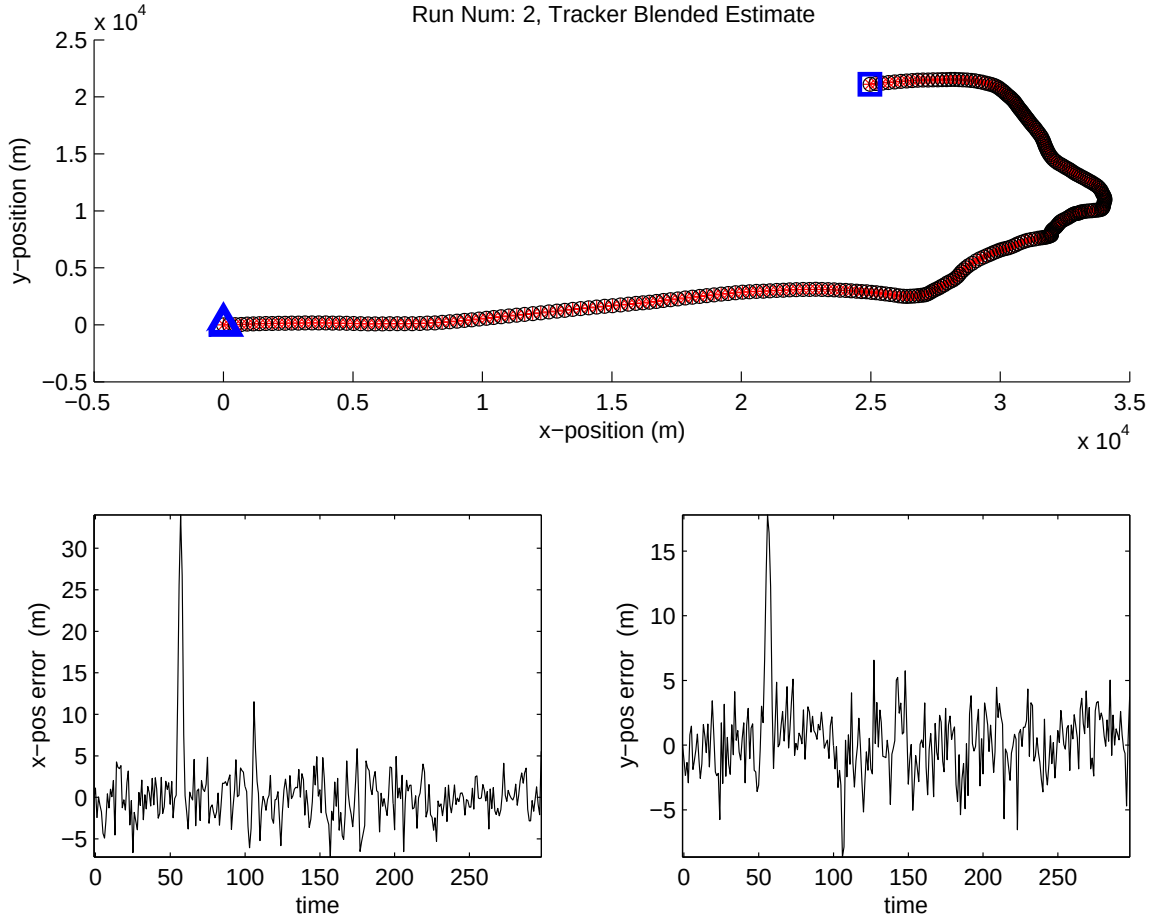

Figure 4.115: Blended solution on Monte Carlo run number 2. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

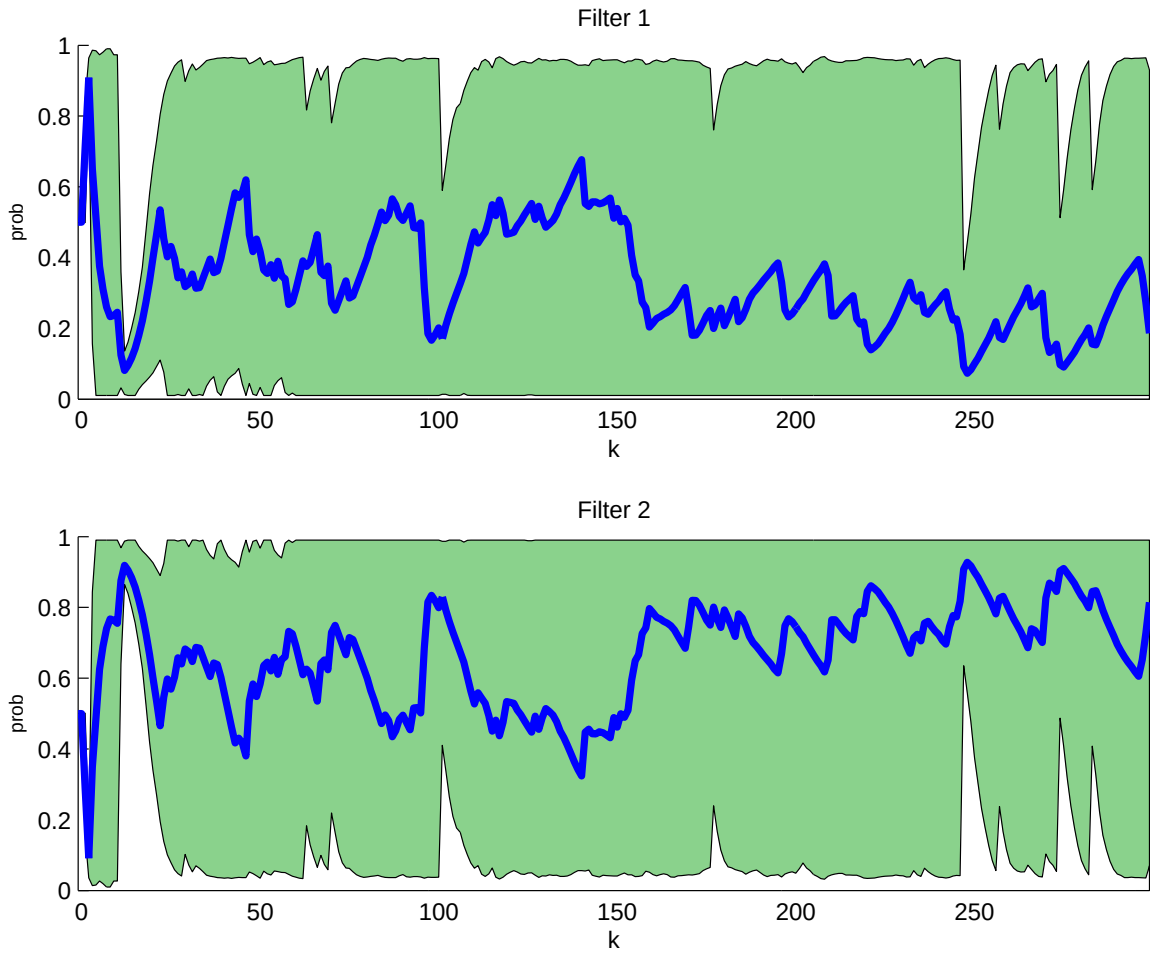

Figure 4.116: Probability flows for the medium clutter density case. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

The summary plots for the IMM appear in Figs. 4.116 through 4.119. Notice the blended estimate is extremely good for an MHT configuration. It should be noted that this is *better* than the same filter suite running against the same truth scenario for the IMM operating in the low-clutter environment (see Fig. 4.64). The performance seen here will be discussed in more detail in this chapter's summary. As stated previously, the IMM simulation data reflects the full set of 10 Monte Carlo trials.

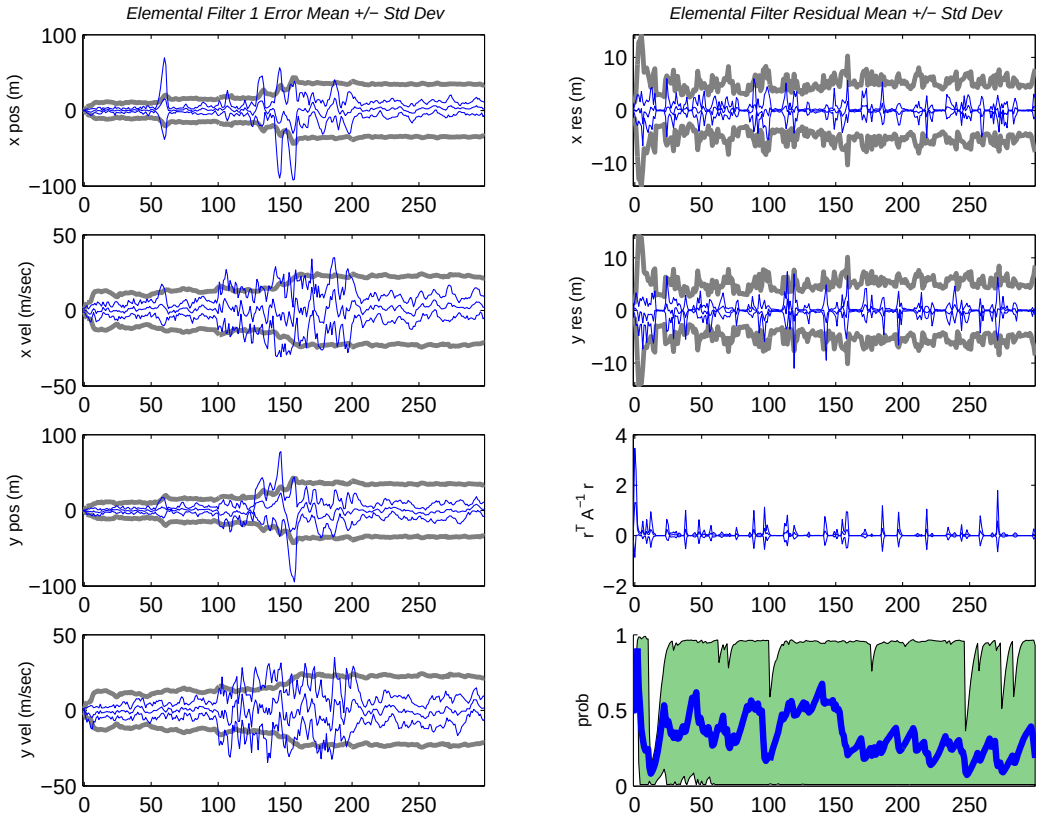

Figure 4.117: Filter 1 data for 10 Monte Carlo runs. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

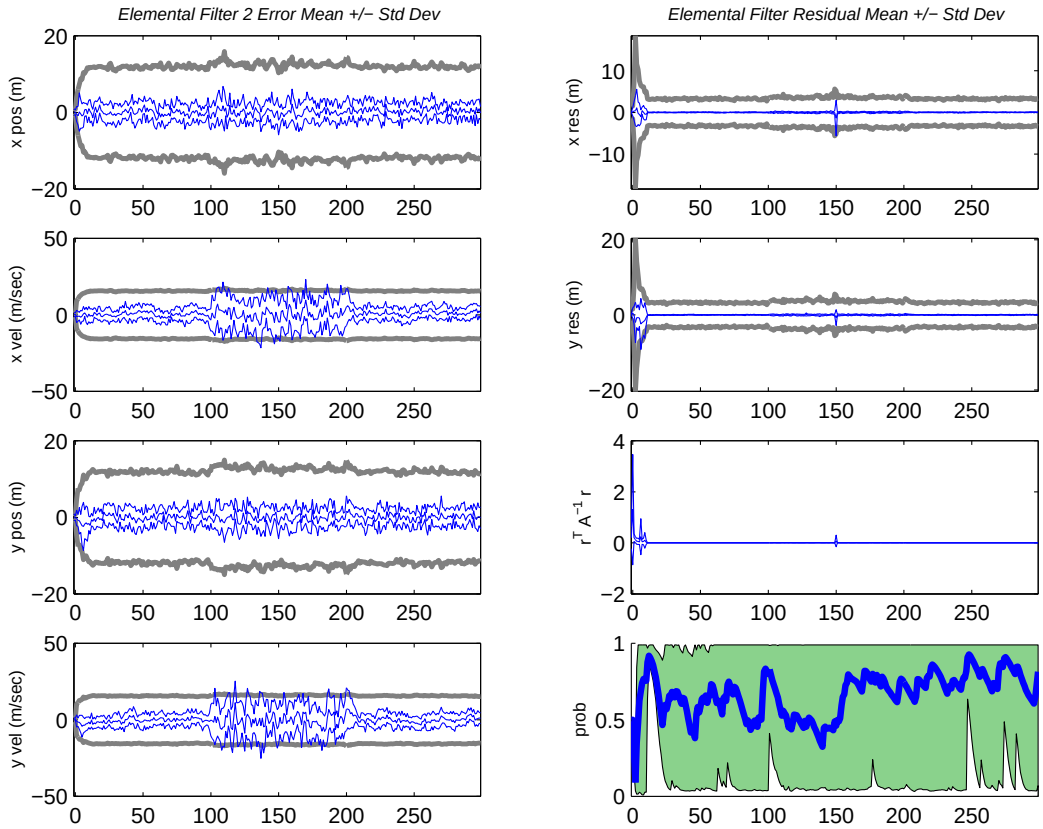

Figure 4.118: Filter 2 data for 10 Monte Carlo runs. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

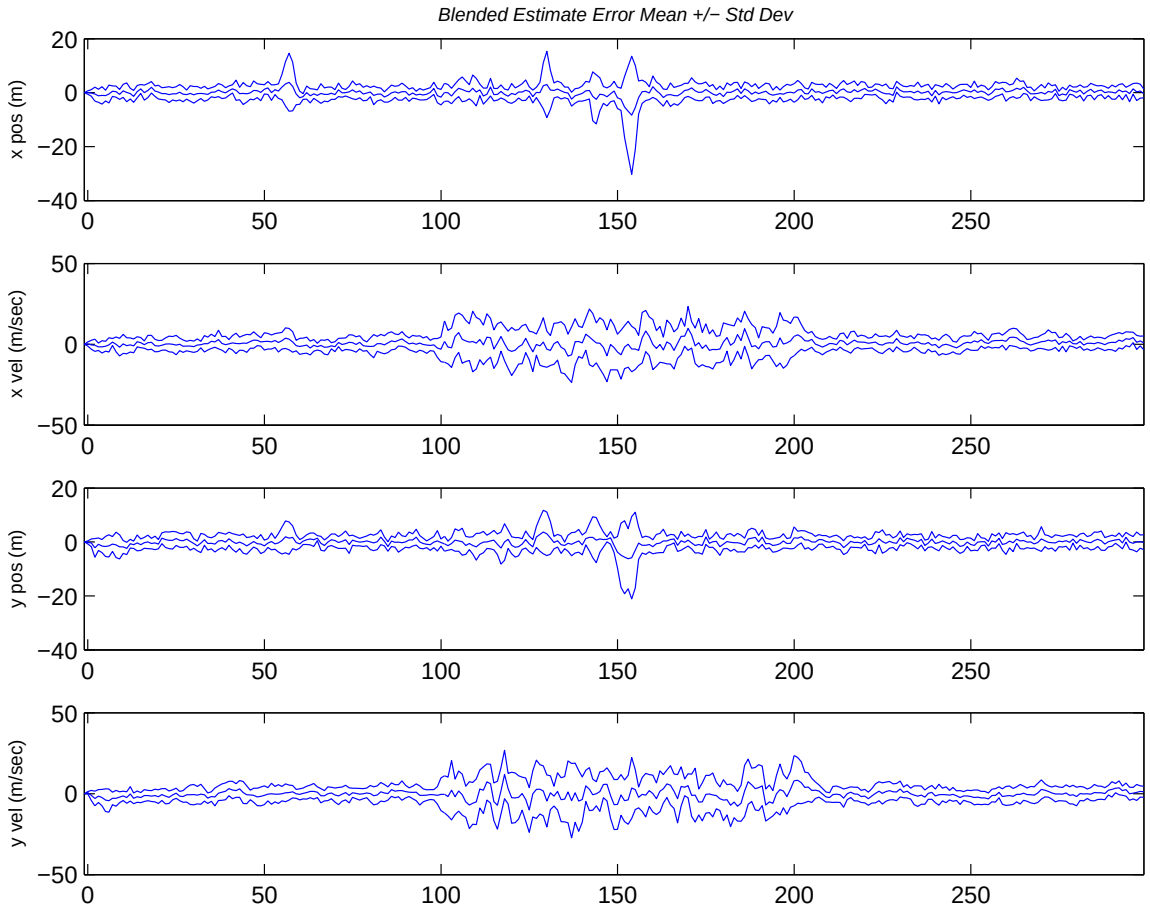

Figure 4.119: Blended estimate for the medium clutter density case. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

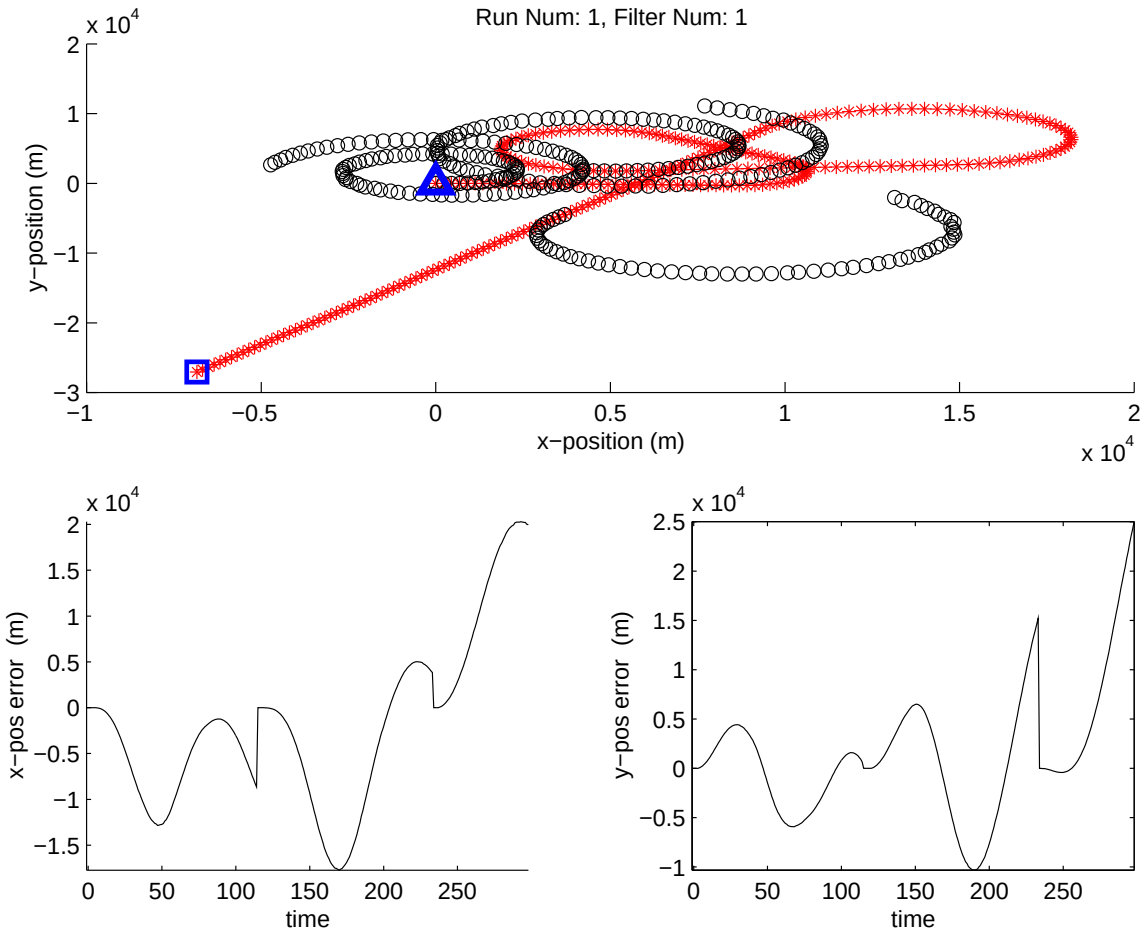

Figure 4.120: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

4.5.2 Suite 5 vs. Scenario 5. The single-run MMAE results (run number 1) for filter Suite 5 running against truth Scenario 5 appear in Figs. 4.120 through 4.123. The single-run performance is not altogether different from that seen in the low-clutter cases in Figs. 4.68 through 4.71. Note that, unlike the other medium-clutter cases, this Monte Carlo trial number 1 has an identical truth trajectory to the equivalent low clutter case.

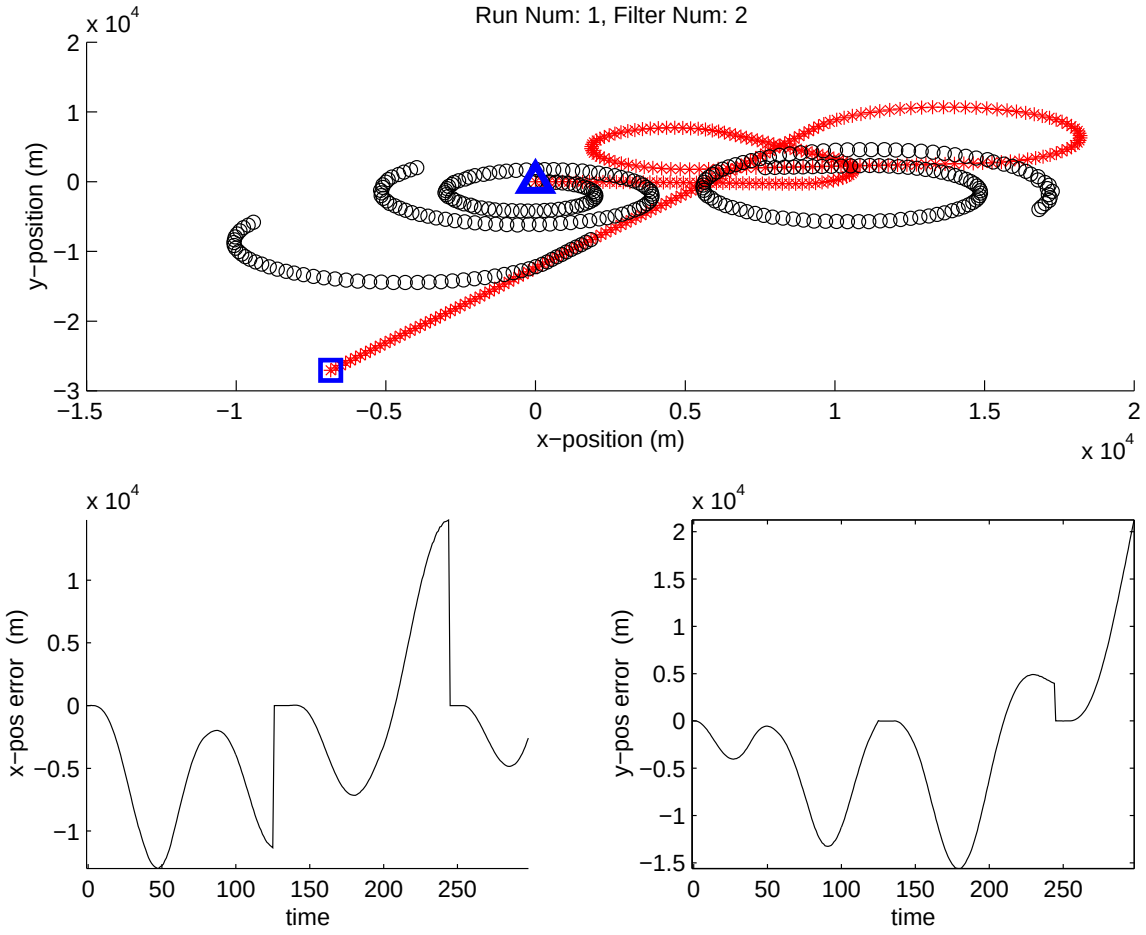

Figure 4.121: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

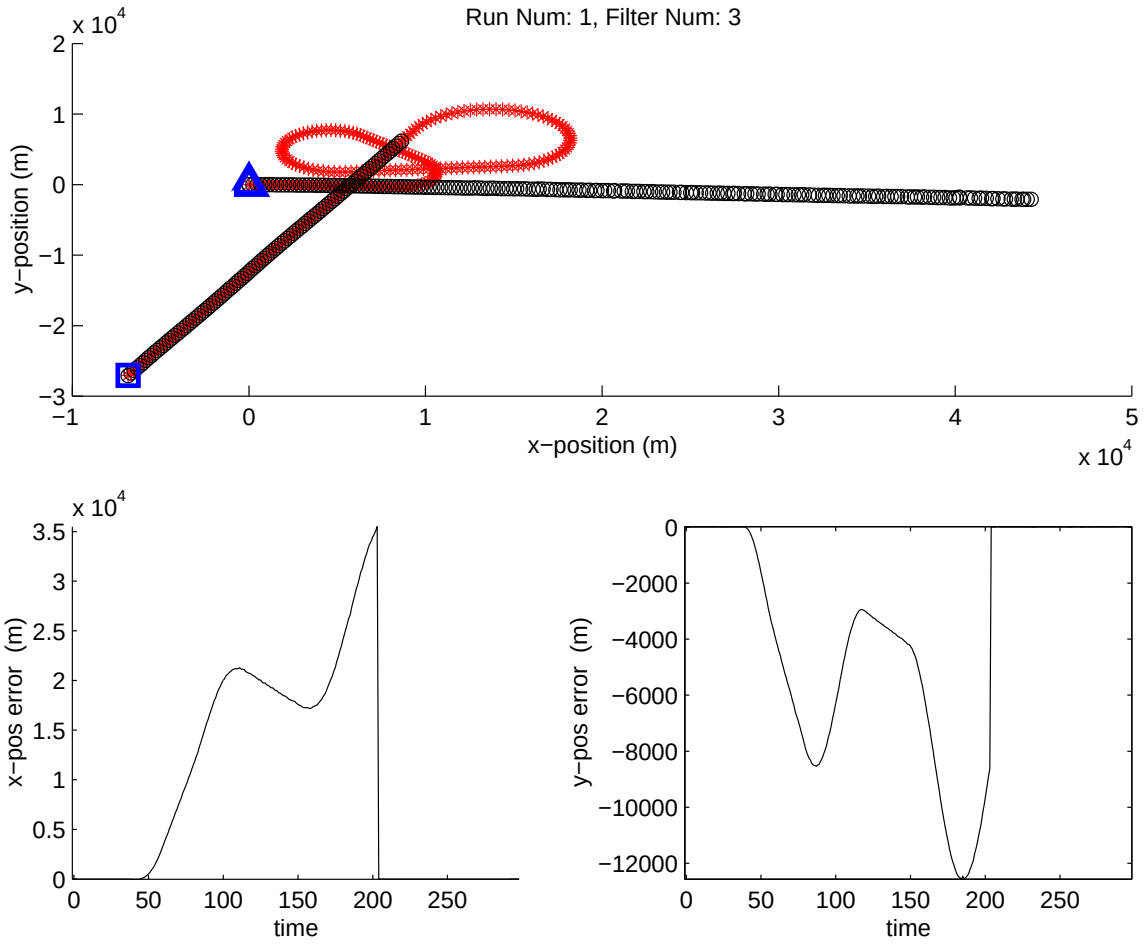

Figure 4.122: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

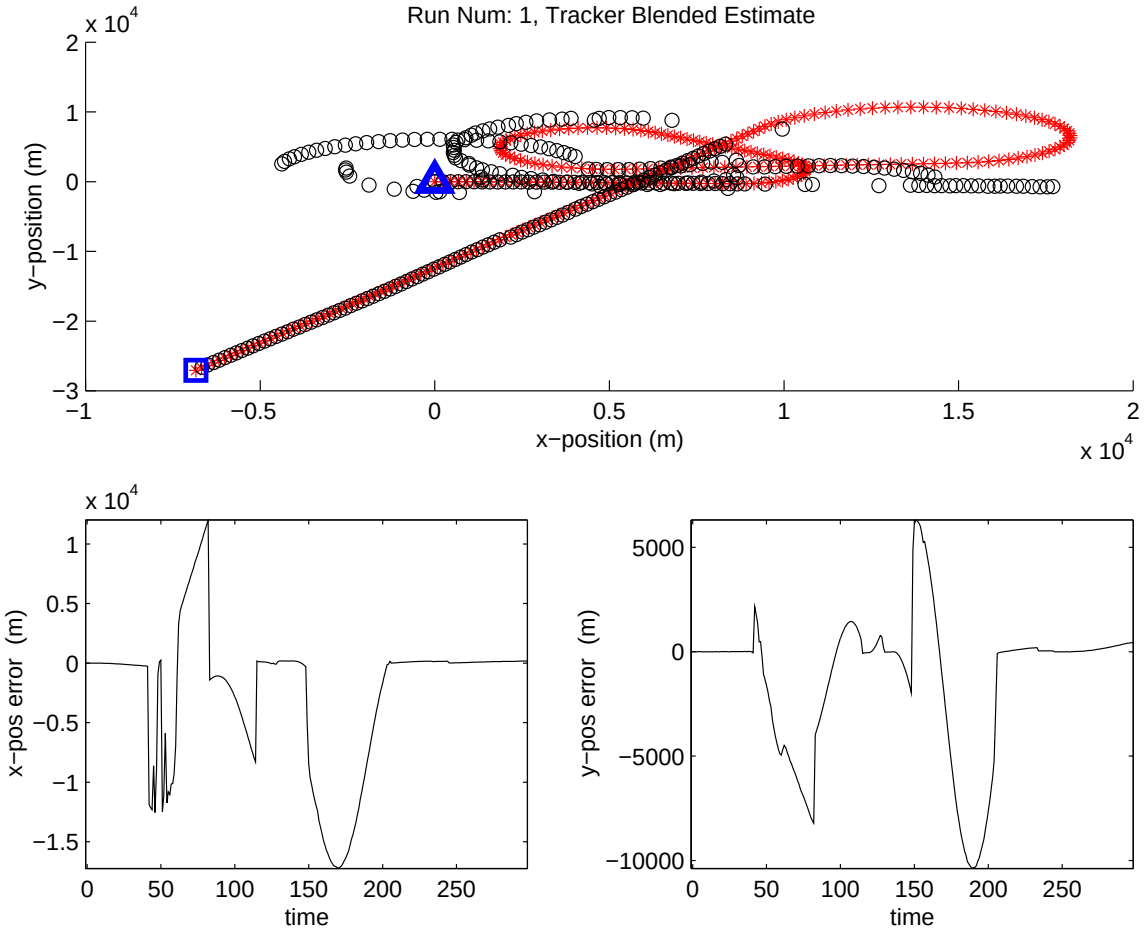

Figure 4.123: Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 5 vs. Scenario 5: MMAE)

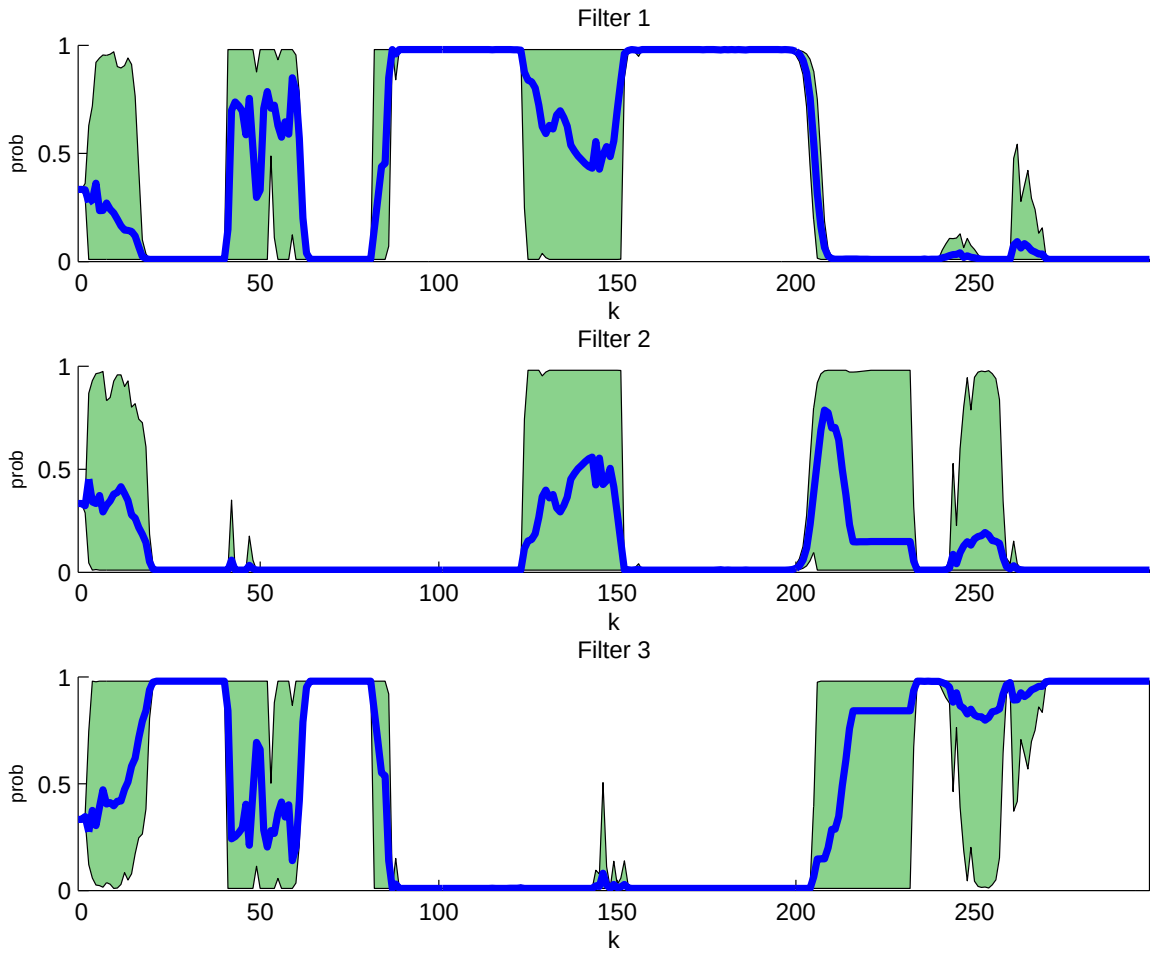

Figure 4.124: Probability flows for the medium clutter density case. (Suite 5 vs. Scenario 5: MMAE)

The summary plots for MMAE appear in Figs. 4.124 through 4.128. Note the probability flows are similar to those seen in the low-clutter case (Fig. 4.72), although the medium-clutter TPV filter number 1 does obtain the probability weight as well as the low-clutter filter during the truth model TPV maneuvers. Notice that the blended estimate performance *improves* in the medium-clutter case (compare to Fig. 4.76), exhibiting a tighter envelope and smaller magnitude excursions in the $\pm 1\sigma$ error plots. There are no IMM results for this case, and only 7 Monte Carlo trials are reflected in the MMAE simulation data (time limitations prevented the completion of the final three trials).

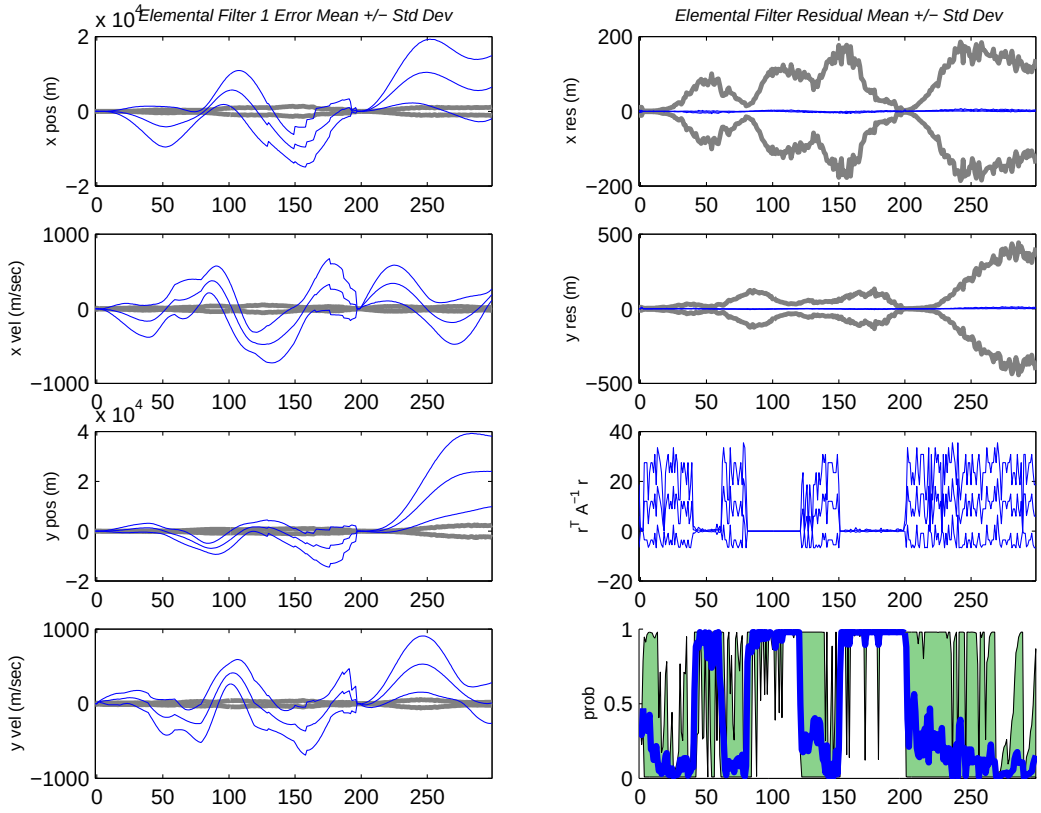

Figure 4.125: Filter 1 data for 7 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

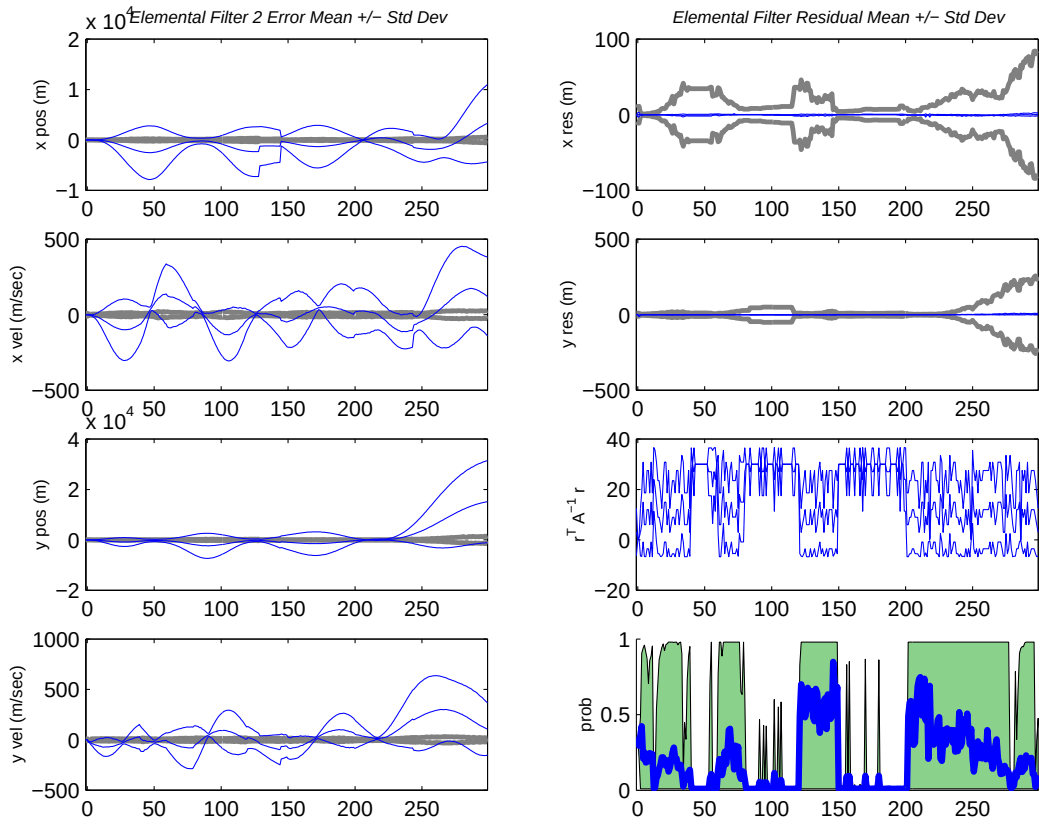

Figure 4.126: Filter 2 data for 7 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

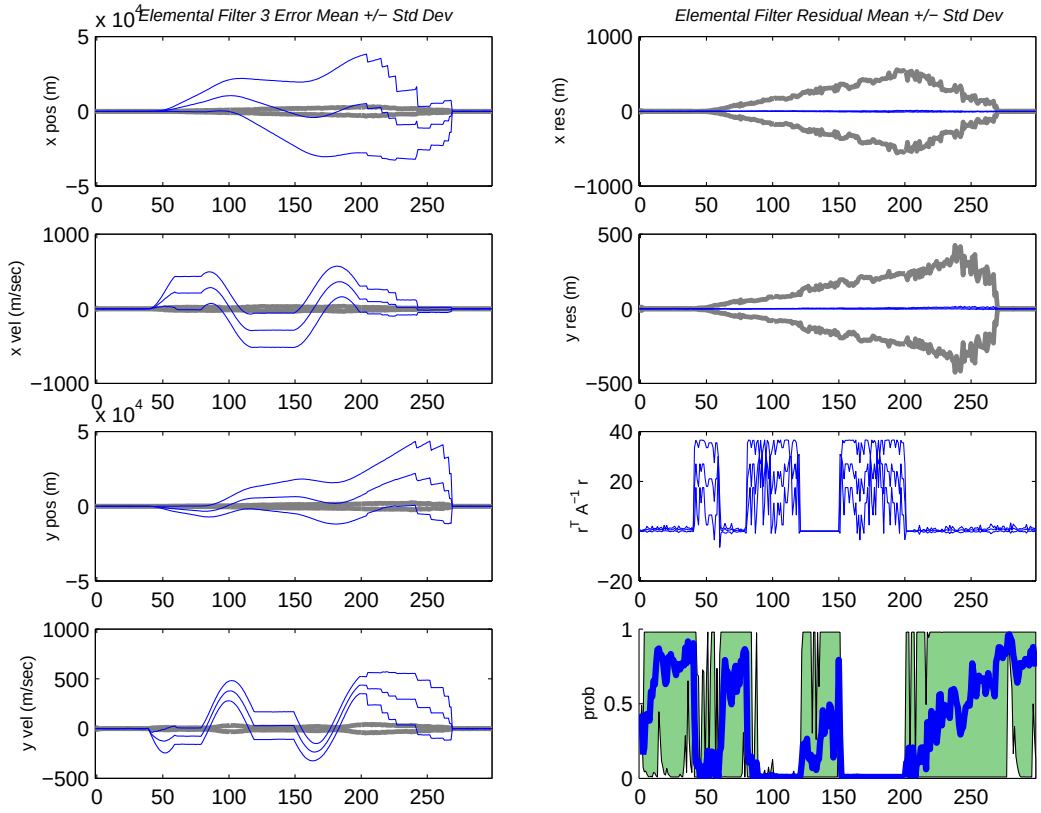

Figure 4.127: Filter 3 data for 7 Monte Carlo runs. (Suite 5 vs. Scenario 5: MMAE)

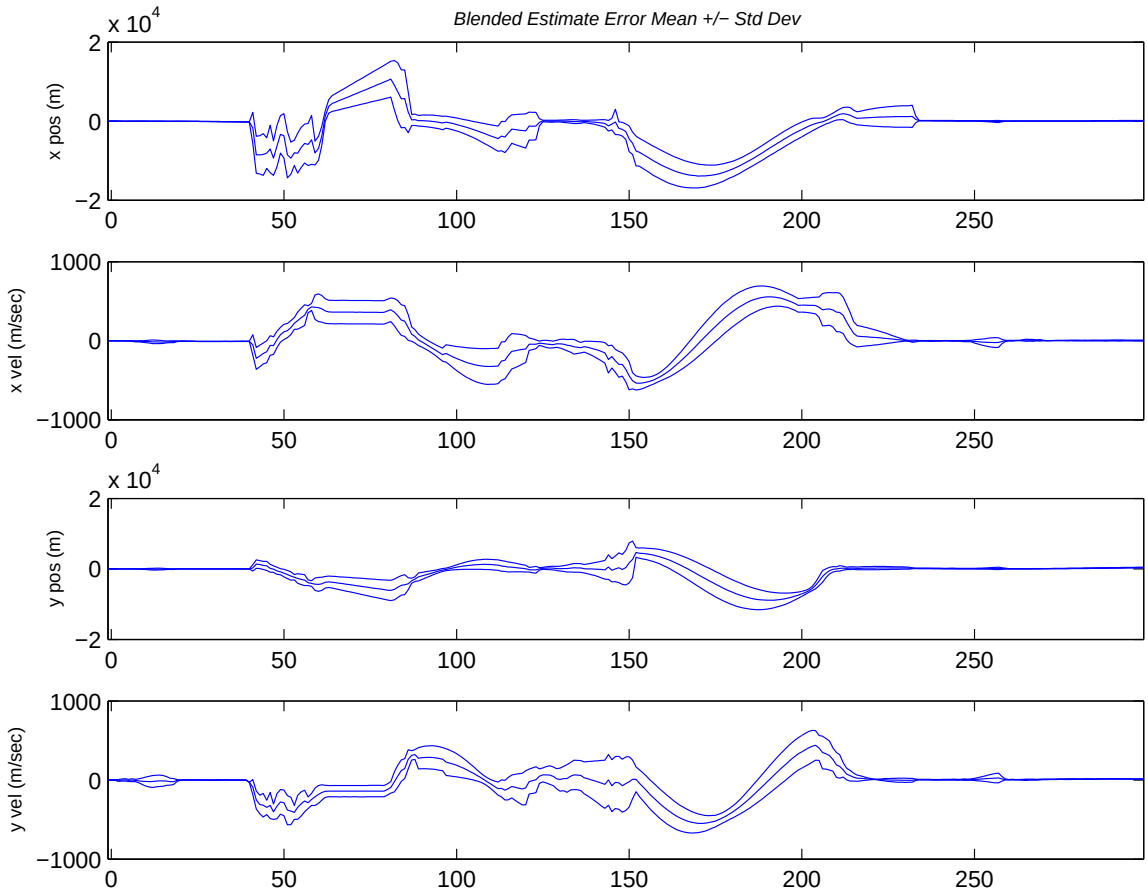

Figure 4.128: Blended estimate for the medium clutter density case. (Suite 5 vs. Scenario 5: MMAE)

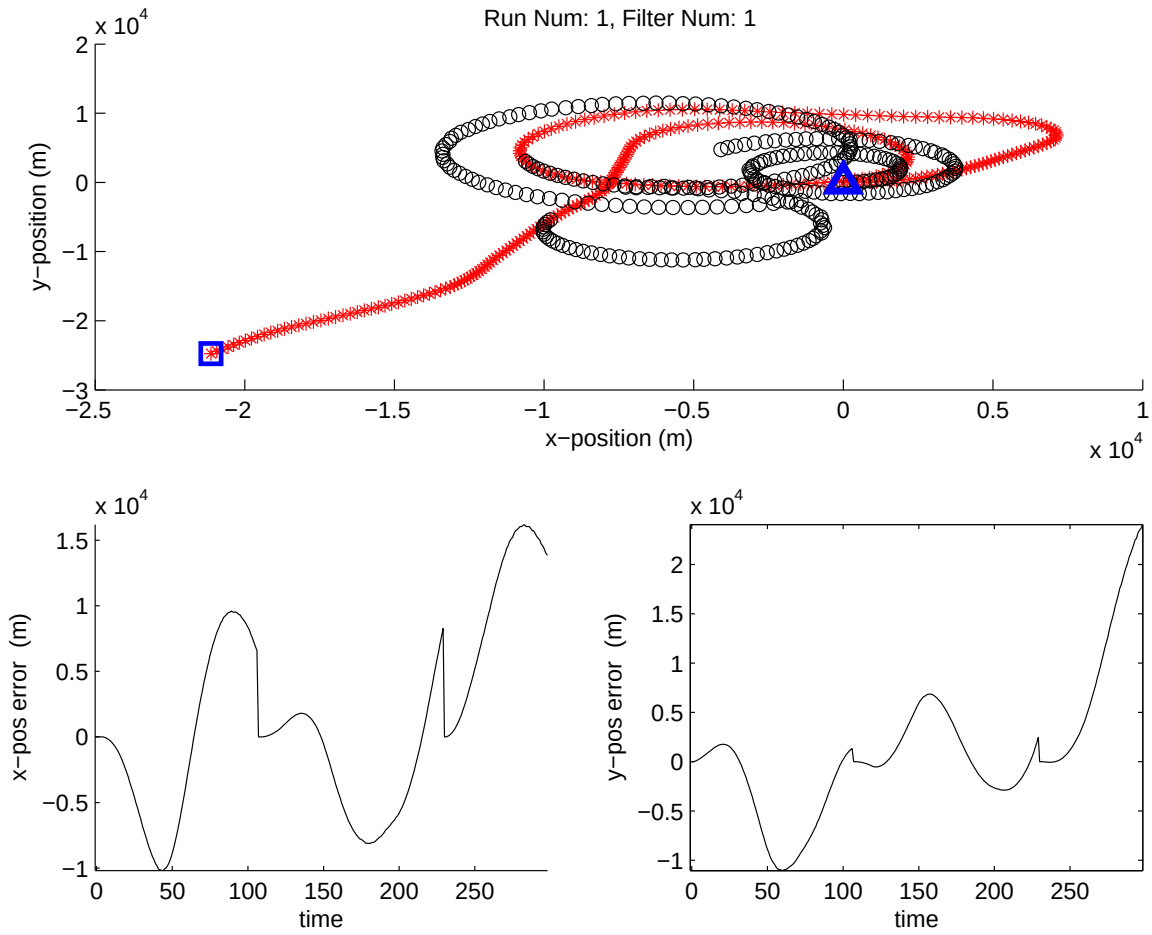

Figure 4.129: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

4.5.3 Suite 6 vs. Scenario 6. Figures 4.129 through 4.132 show the single-run plots for the MMAE with filter Suite 6 running against truth Scenario 6. Overall, the behavior is very similar to that in the low-clutter case (shown with different truth trajectory in Figs. 4.86 through 4.89), and the actual error committed by the filters and blended estimate are on the same order of magnitude. Notice the blended estimate continually snaps to the truth trajectory over the course of the simulation.

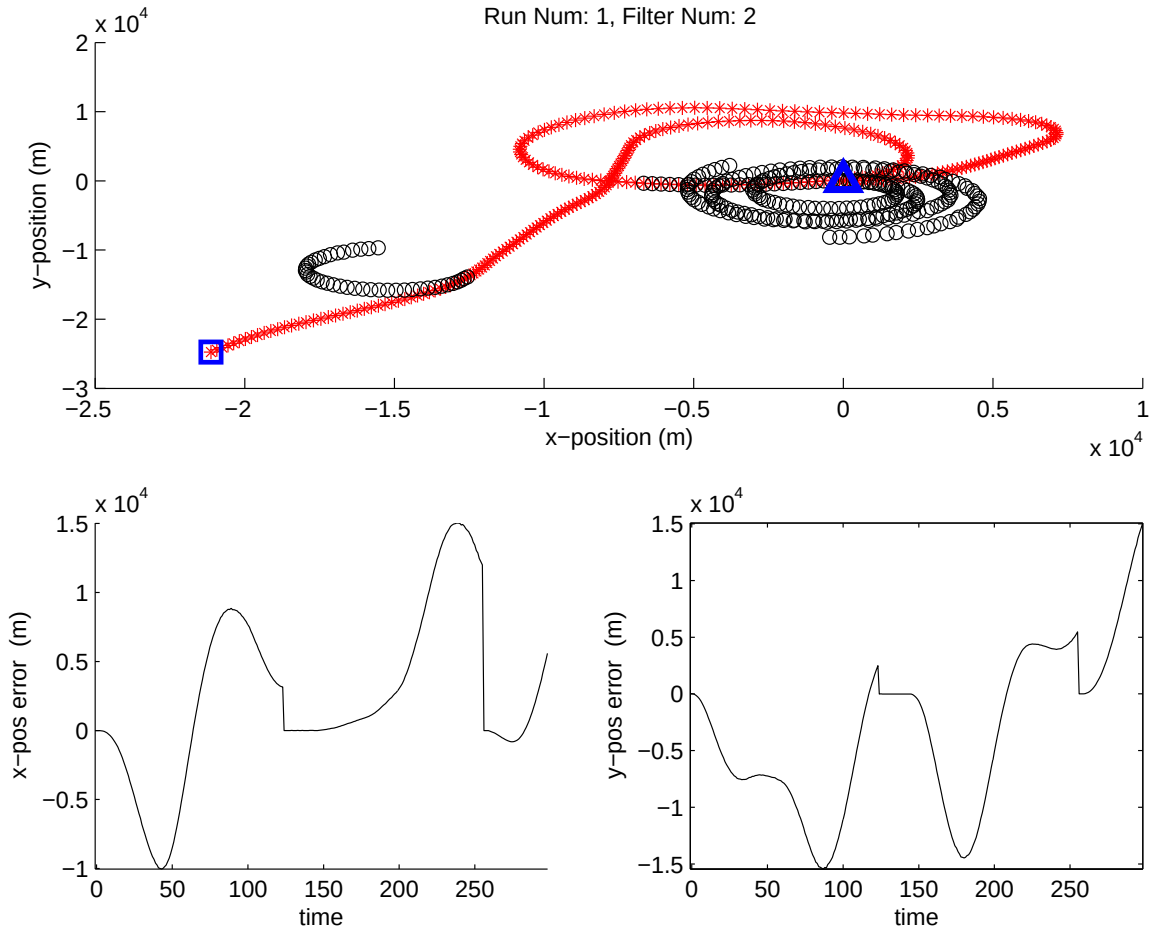

Figure 4.130: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

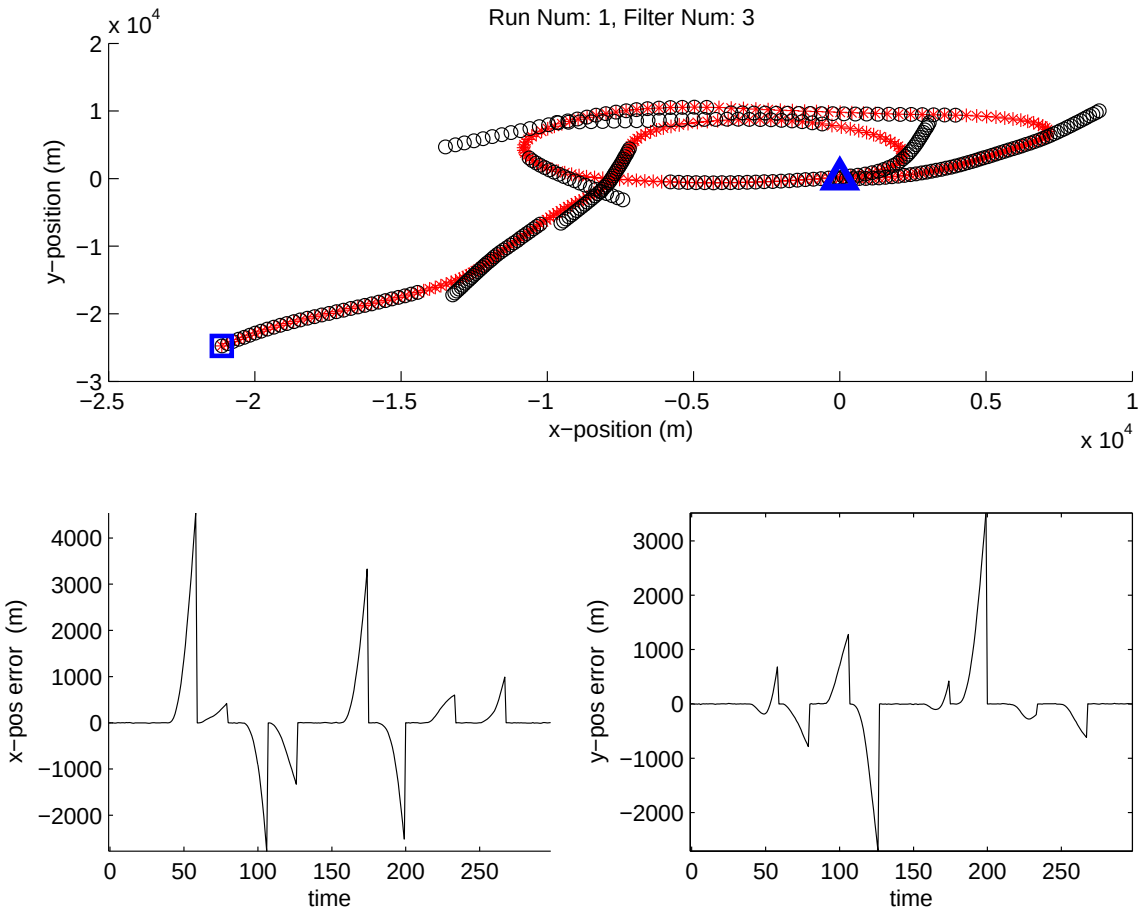

Figure 4.131: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

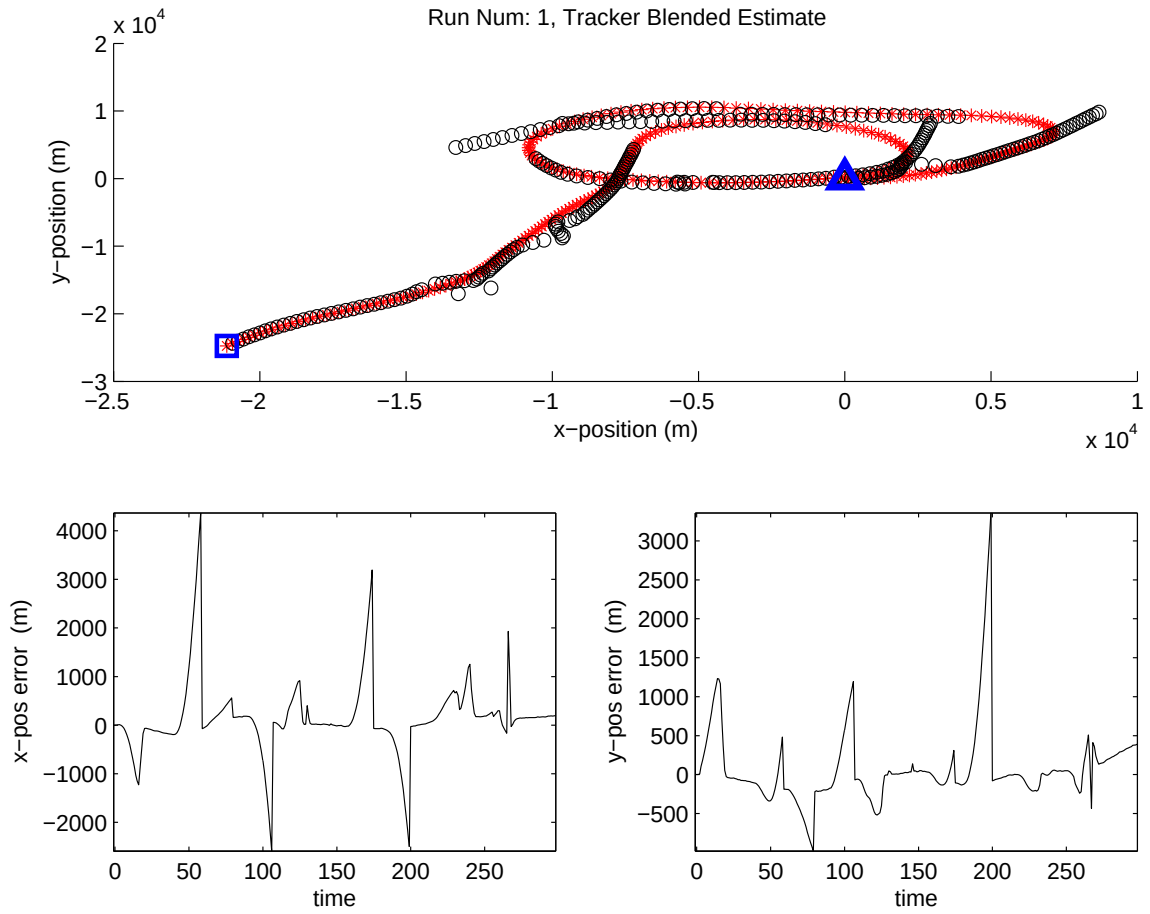

Figure 4.132: Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: MMAE)

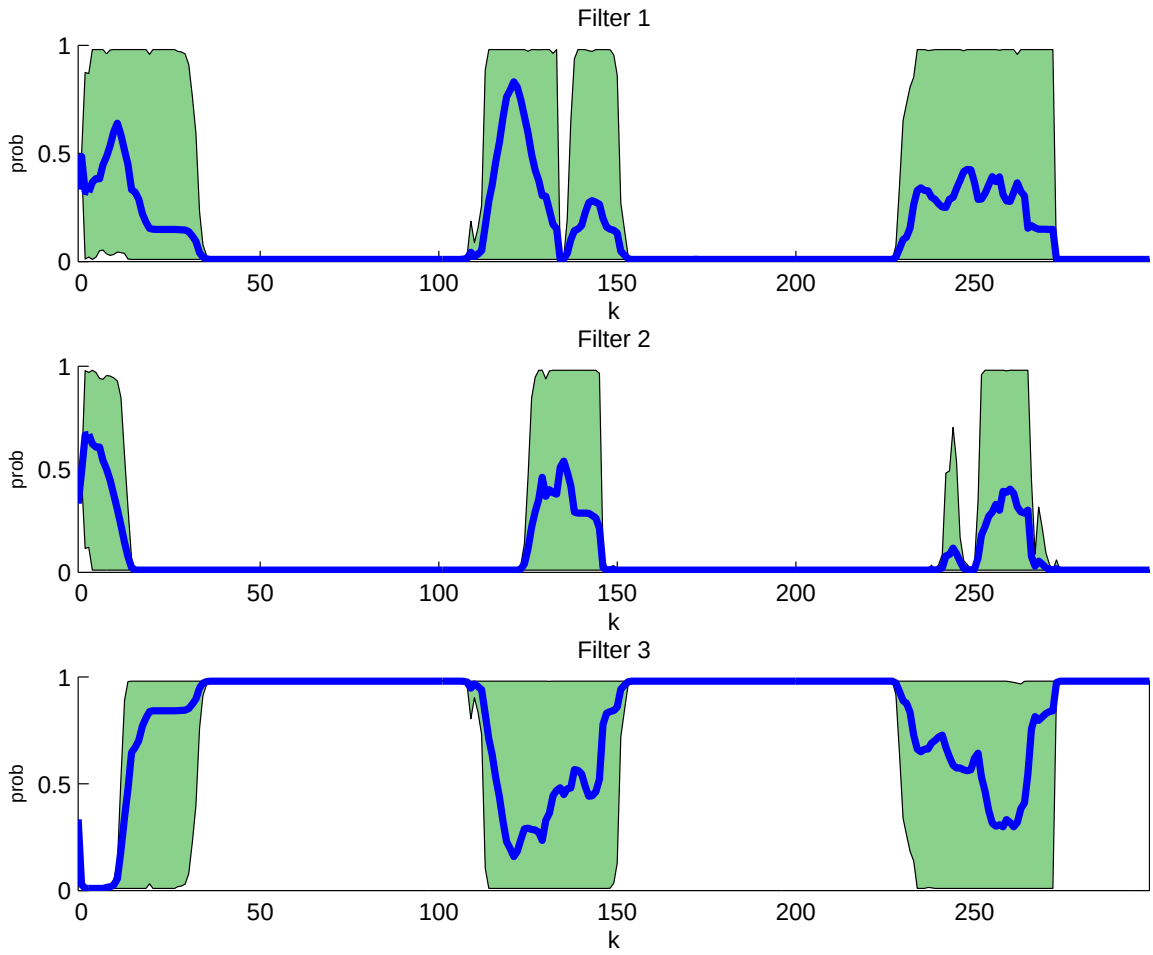

Figure 4.133: Probability flows for the medium clutter density case. (Suite 6 vs. Scenario 6: MMAE)

The summary plots are shown in Figs. 4.133 through 4.137. As was the case for the low-clutter scenario (Fig. 4.91, the probability flow is quite ambiguous and gives little indication of what the truth model is actually doing. Nonetheless, the medium-clutter blended estimate is almost exactly like the low-clutter estimate seen in Fig. 4.95, with actual error committed having similar magnitude mean $\pm 1\sigma$ plots. Note that the MMAE simulation data represents only 7 Monte Carlo trials (time limitations prevented the completion of the final three trials).

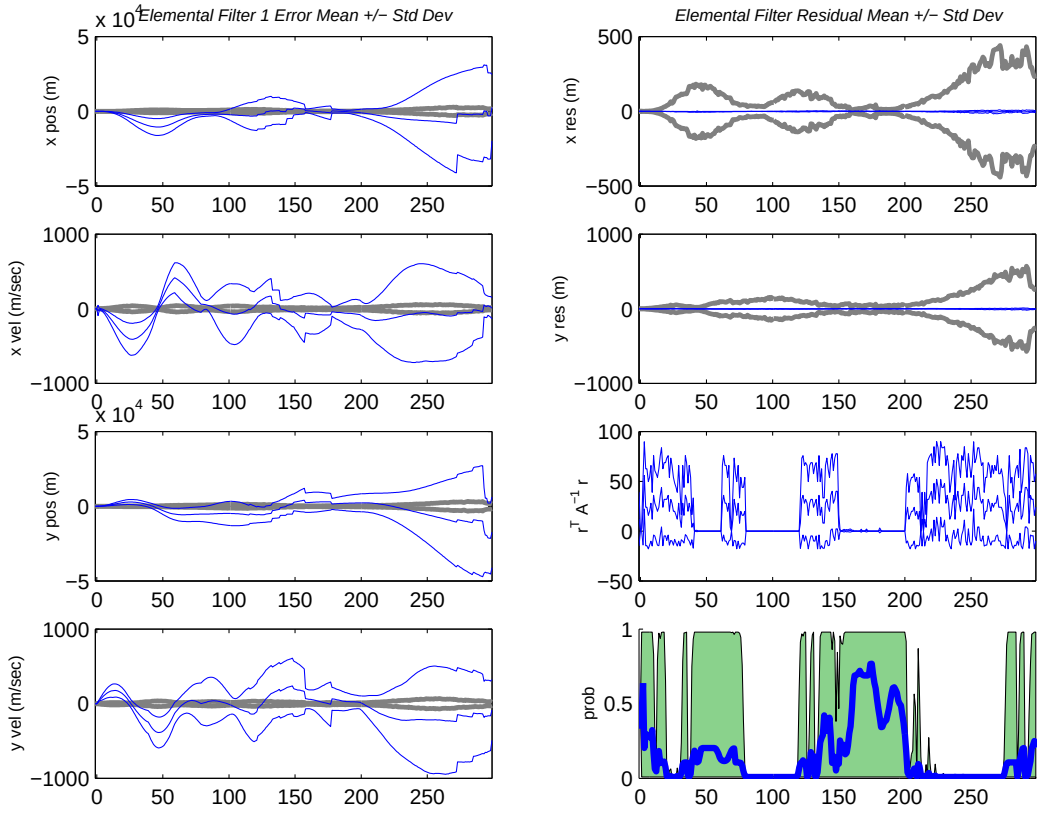

Figure 4.134: Filter 1 data for 7 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

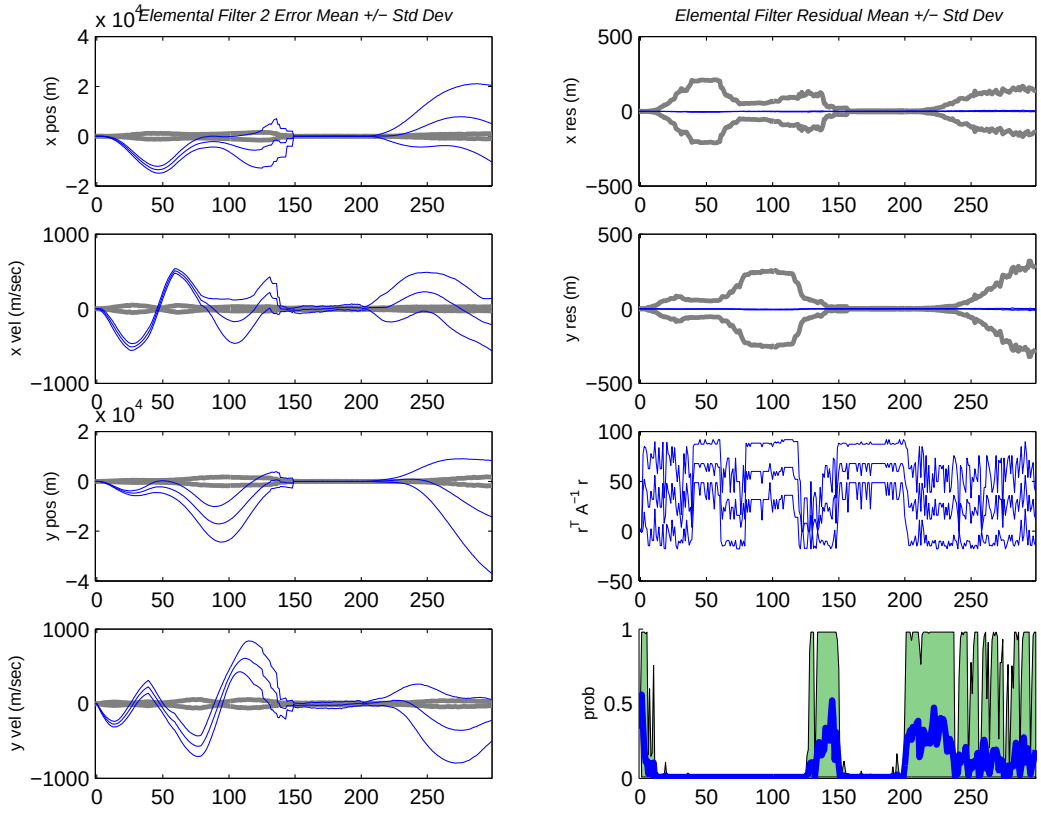

Figure 4.135: Filter 2 data for 7 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

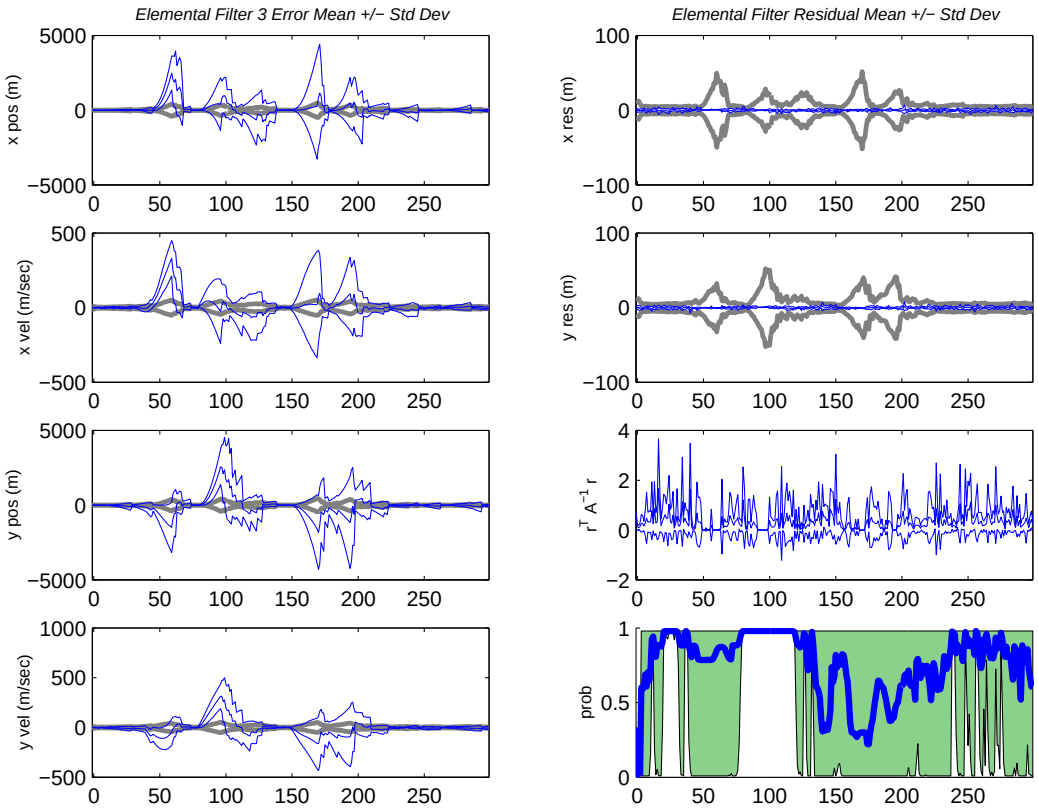

Figure 4.136: Filter 3 data for 7 Monte Carlo runs. (Suite 6 vs. Scenario 6: MMAE)

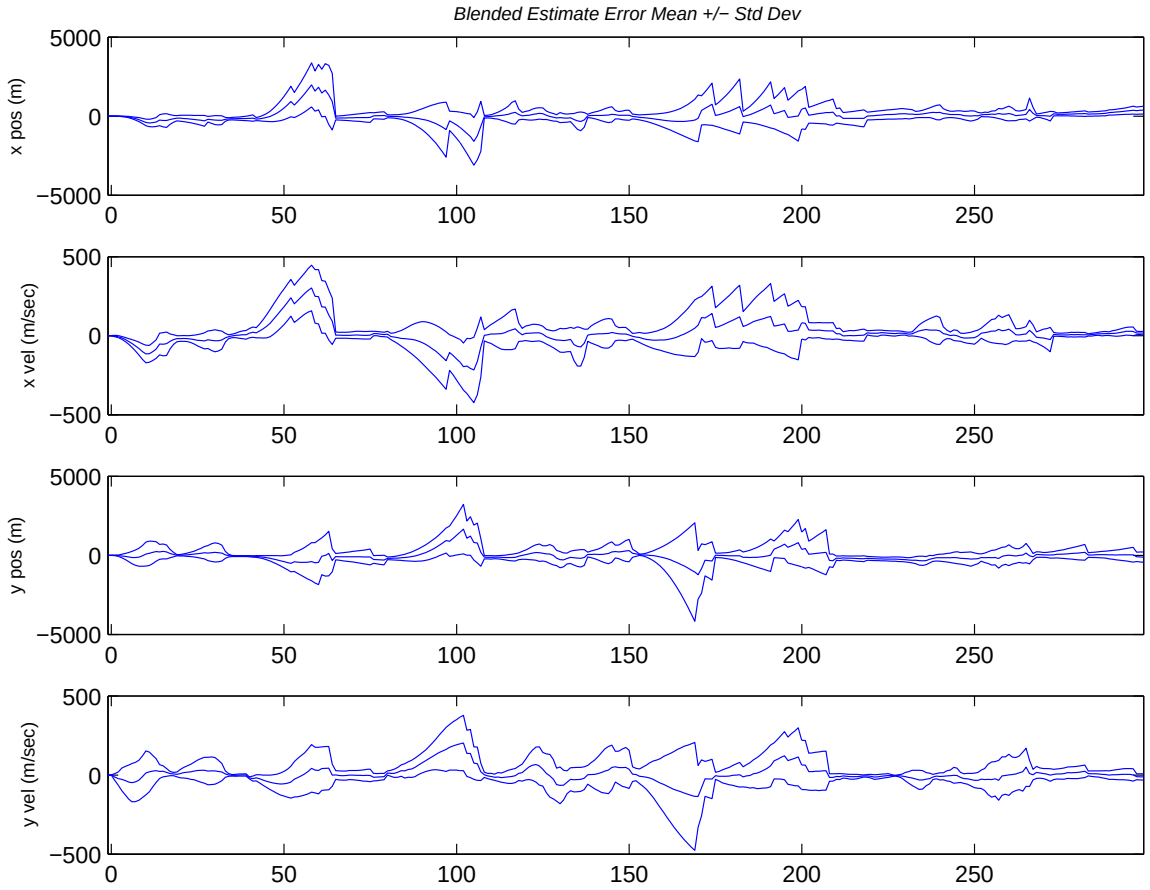

Figure 4.137: Blended estimate for the medium clutter density case. (Suite 6 vs. Scenario 6: MMAE)

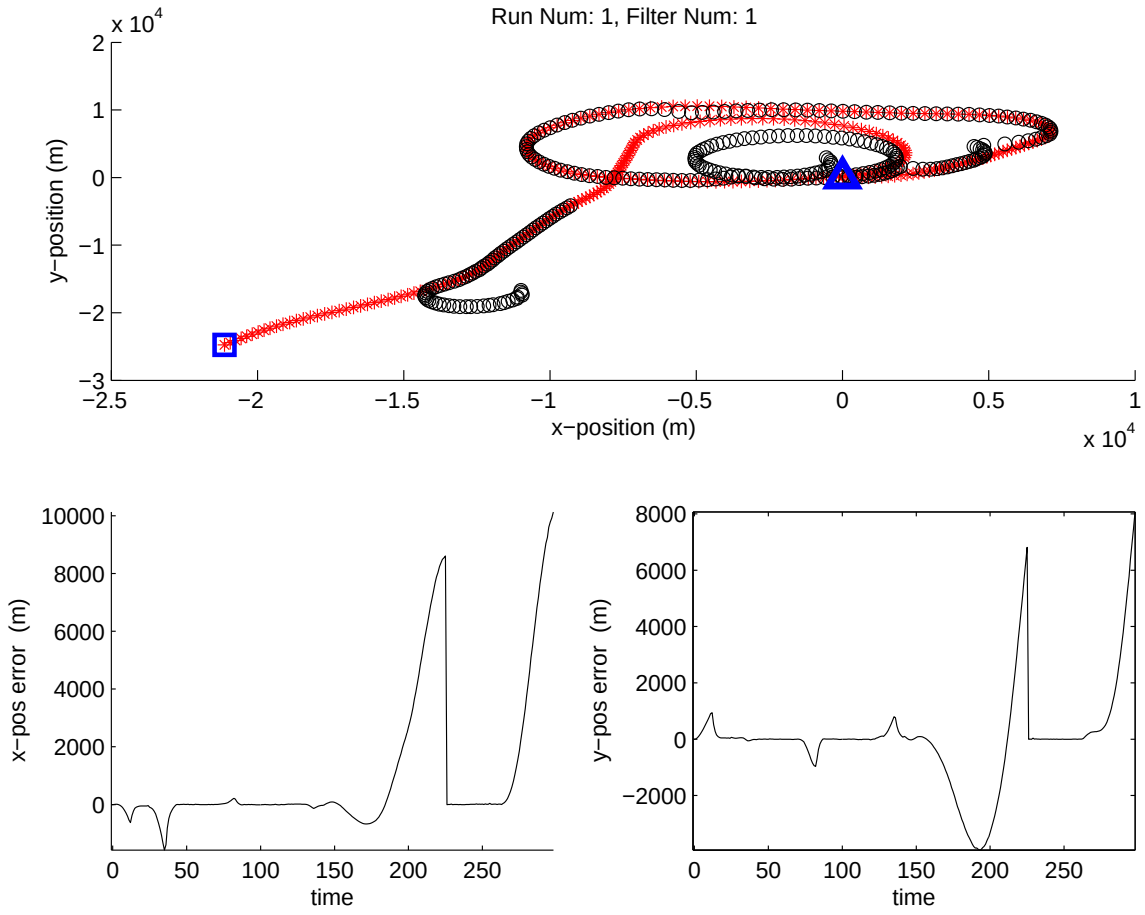

Figure 4.138: Filter 1 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

The run number 1 plots for the IMM appear in Figs. 4.138 through 4.141. Notice the blended estimate on this run is becoming divergent by the end of the simulation. Again, the performance here is on par with that seen in the low-clutter cases (different truth trajectories, Figs. 4.96 through 4.99).

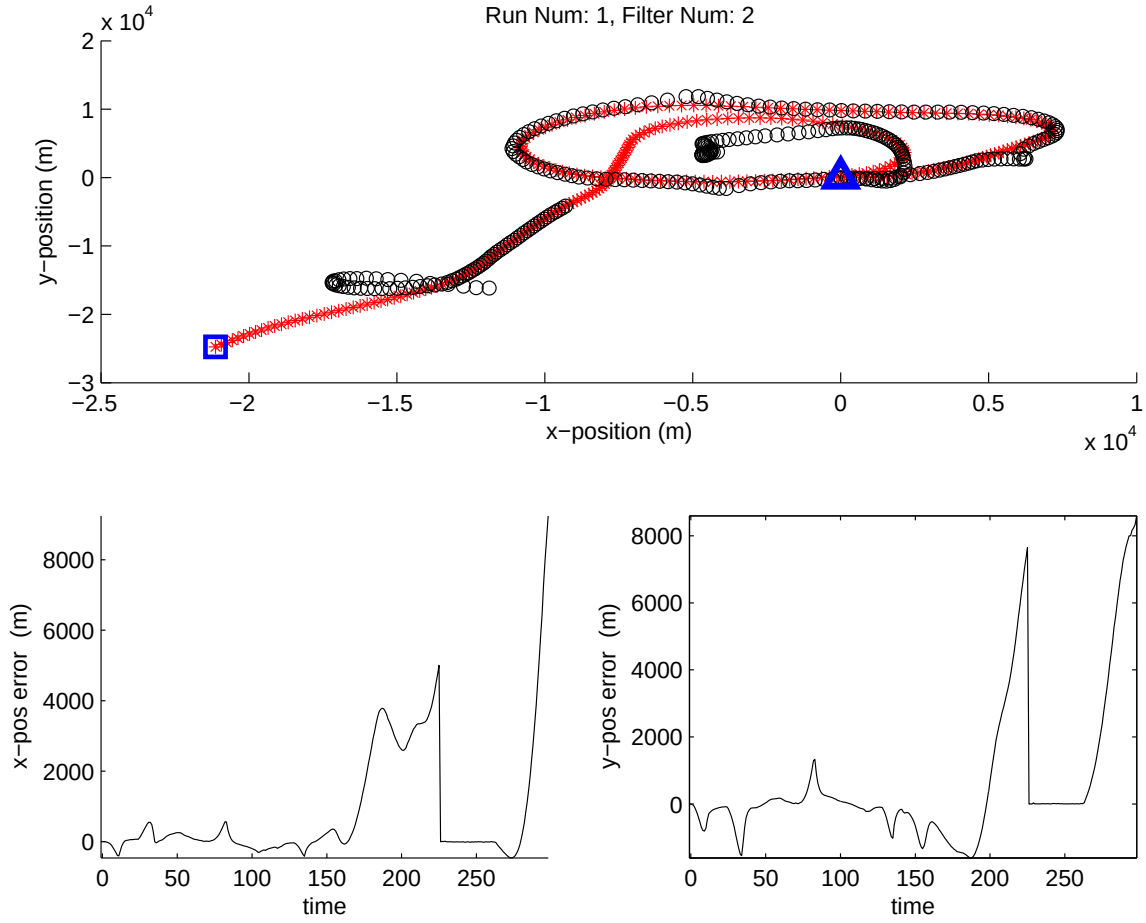

Figure 4.139: Filter 2 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

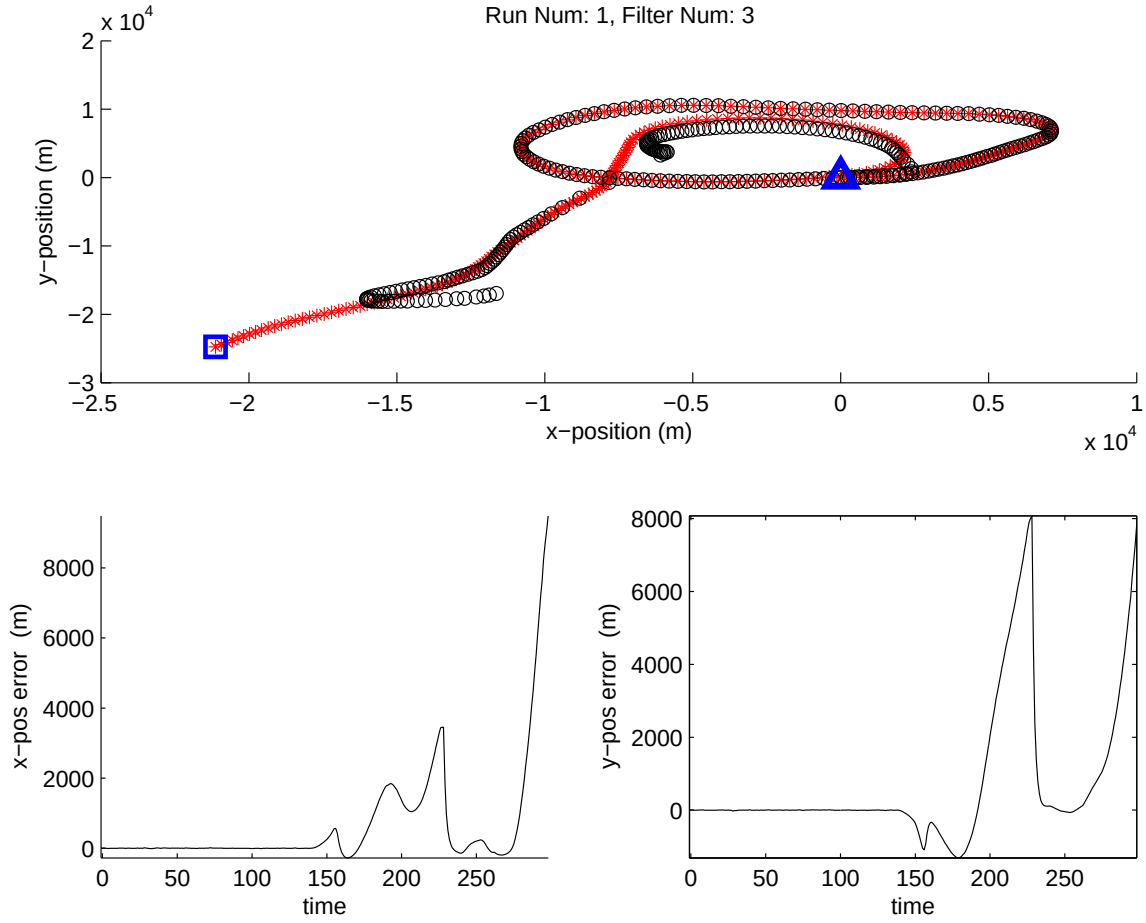

Figure 4.140: Filter 3 estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

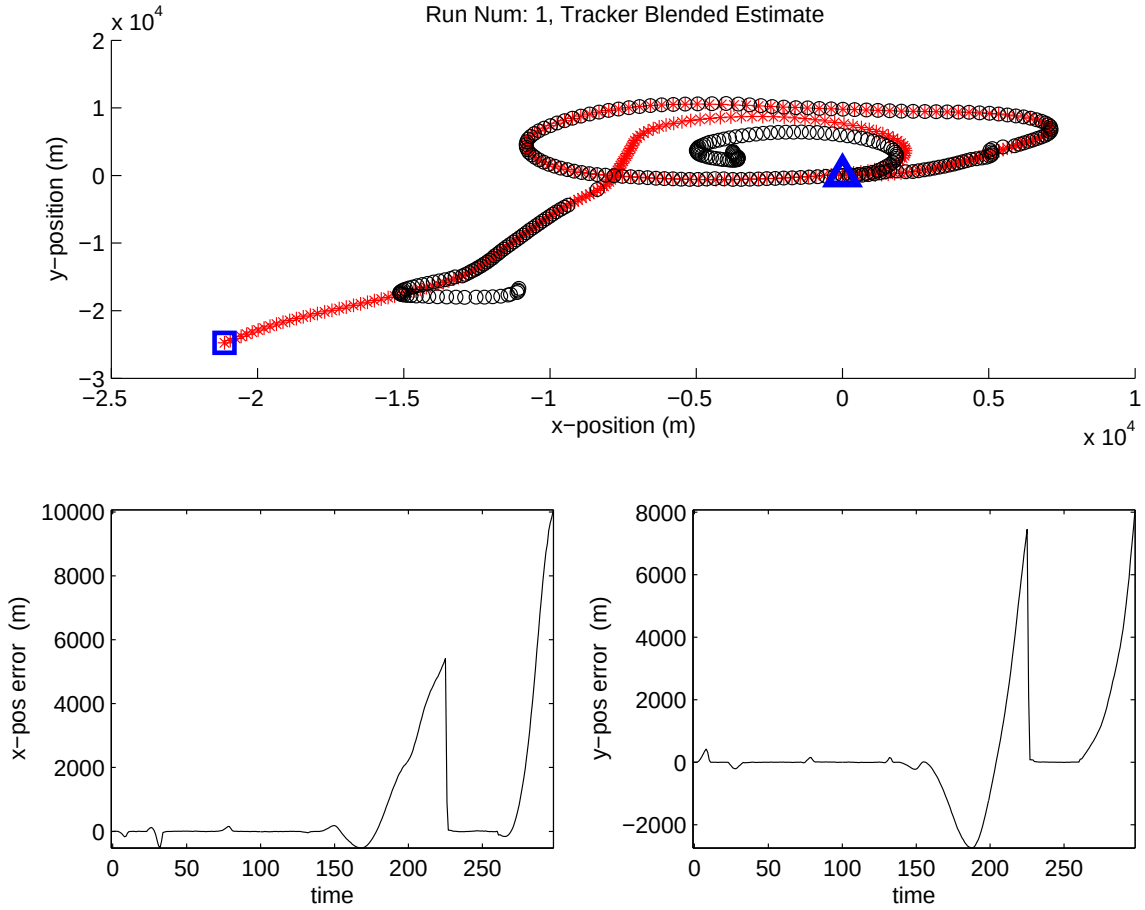

Figure 4.141: IMM Blended estimate for run number 1 in which truth includes three, +3-g TPV maneuvers. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

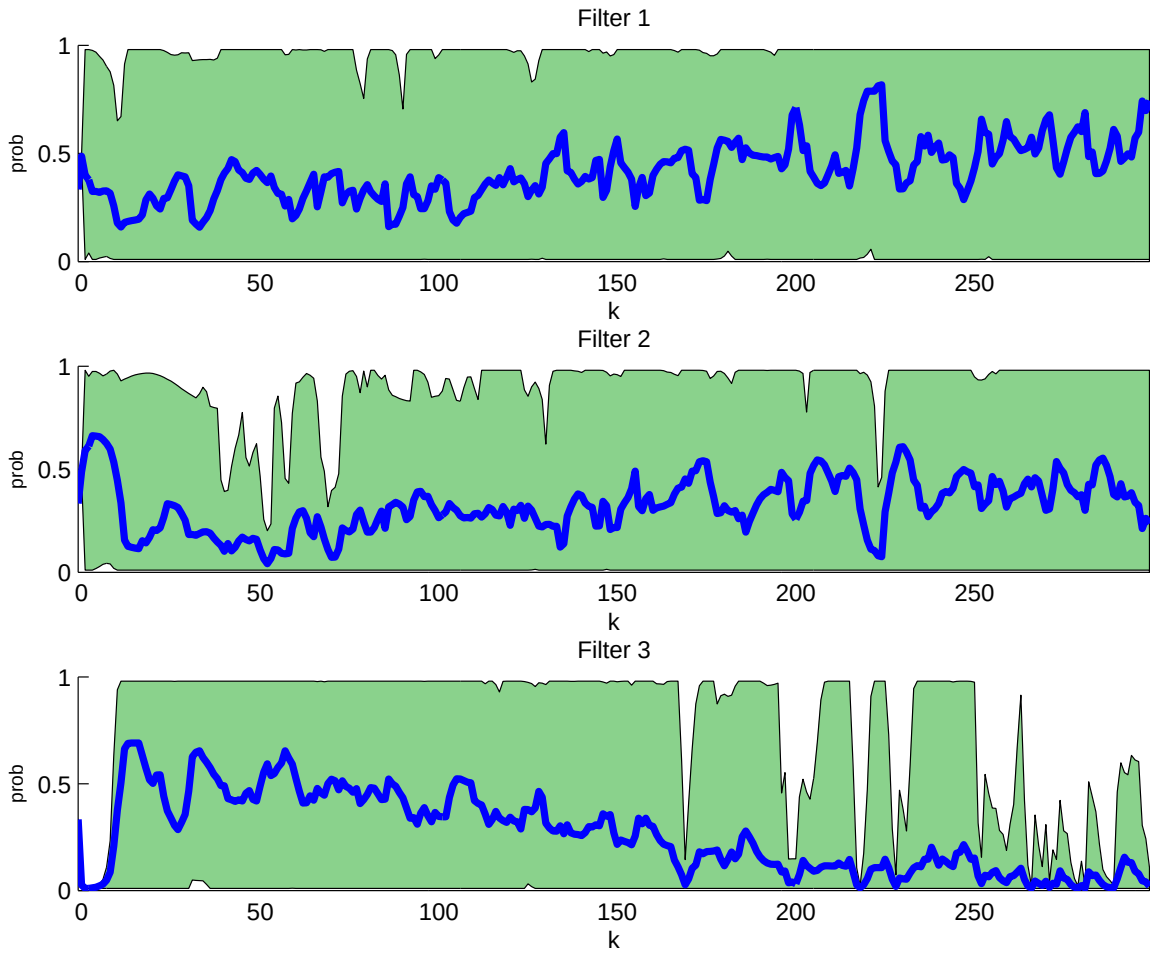

Figure 4.142: Probability flows for the medium clutter density case. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

The summary plots are shown in Figs. 4.142 through 4.146. Note that the probability flow is totally ambiguous, but the blended estimate is quite good. It appears to maintain better tracking of velocity than does the MMAE, and it appears to regain target lock consistently after the cessation of maneuvers (although run number 1, as shown above, may be an exception). Notice the peculiar behavior of the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ plot on filter 3. As stated previously, the pseudo-residual covariance is dominating this computation, as evidenced by the wide gray lines in the residual plots (first and second plot, rightmost column). Note that the IMM simulation data reflects the full set of 10 Monte Carlo trials.

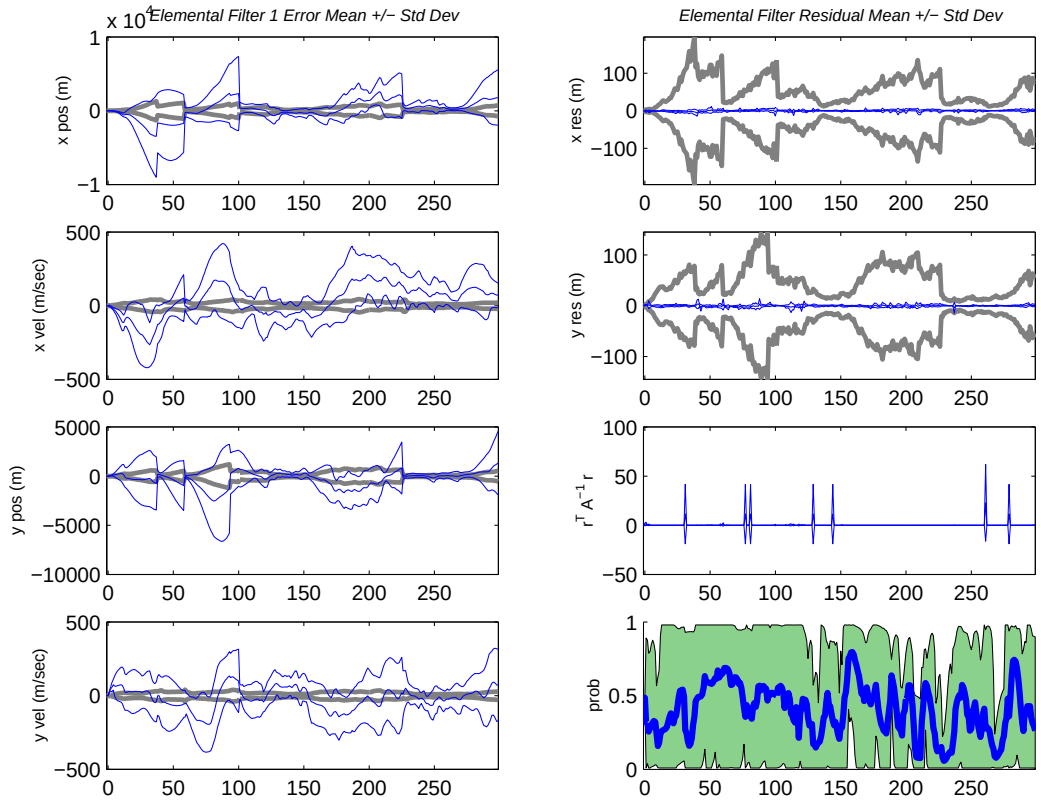

Figure 4.143: Filter 1 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

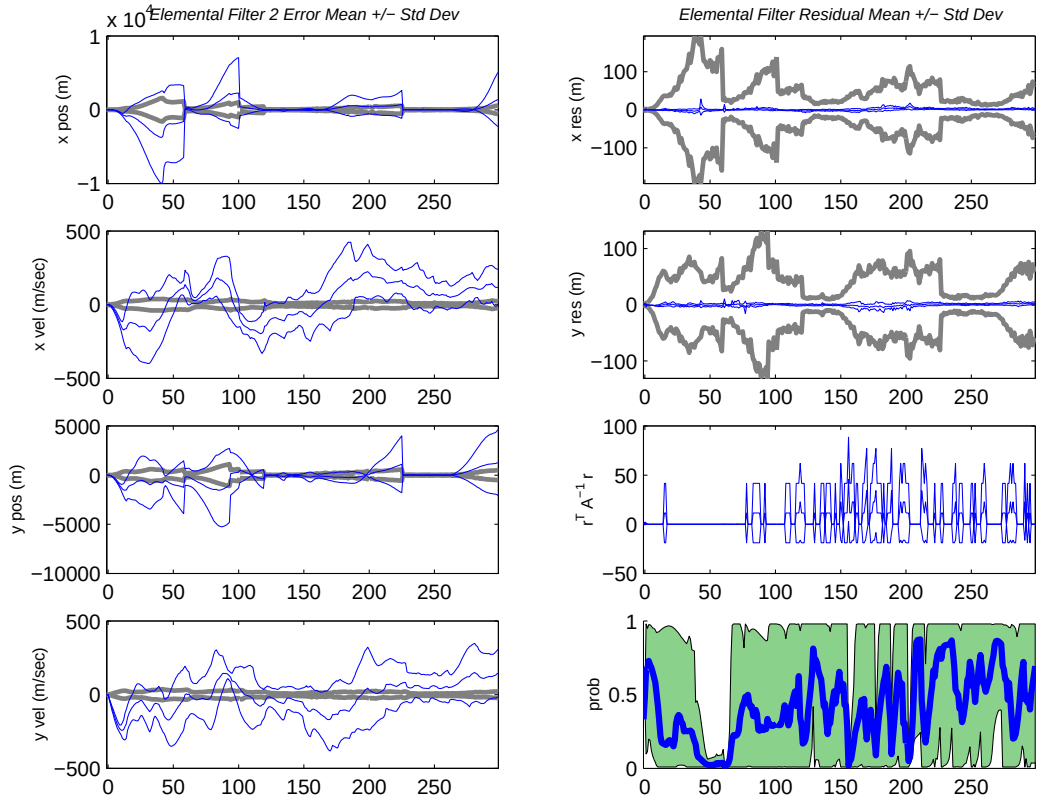

Figure 4.144: Filter 2 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

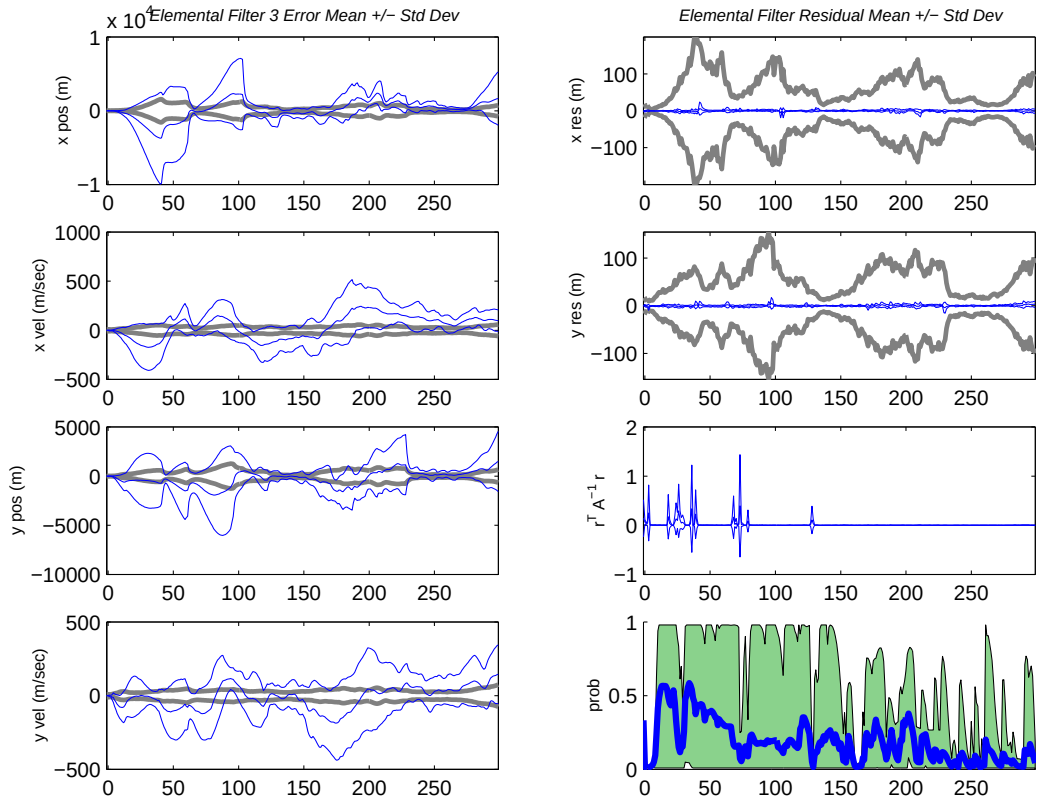

Figure 4.145: Filter 3 data for 10 Monte Carlo runs. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

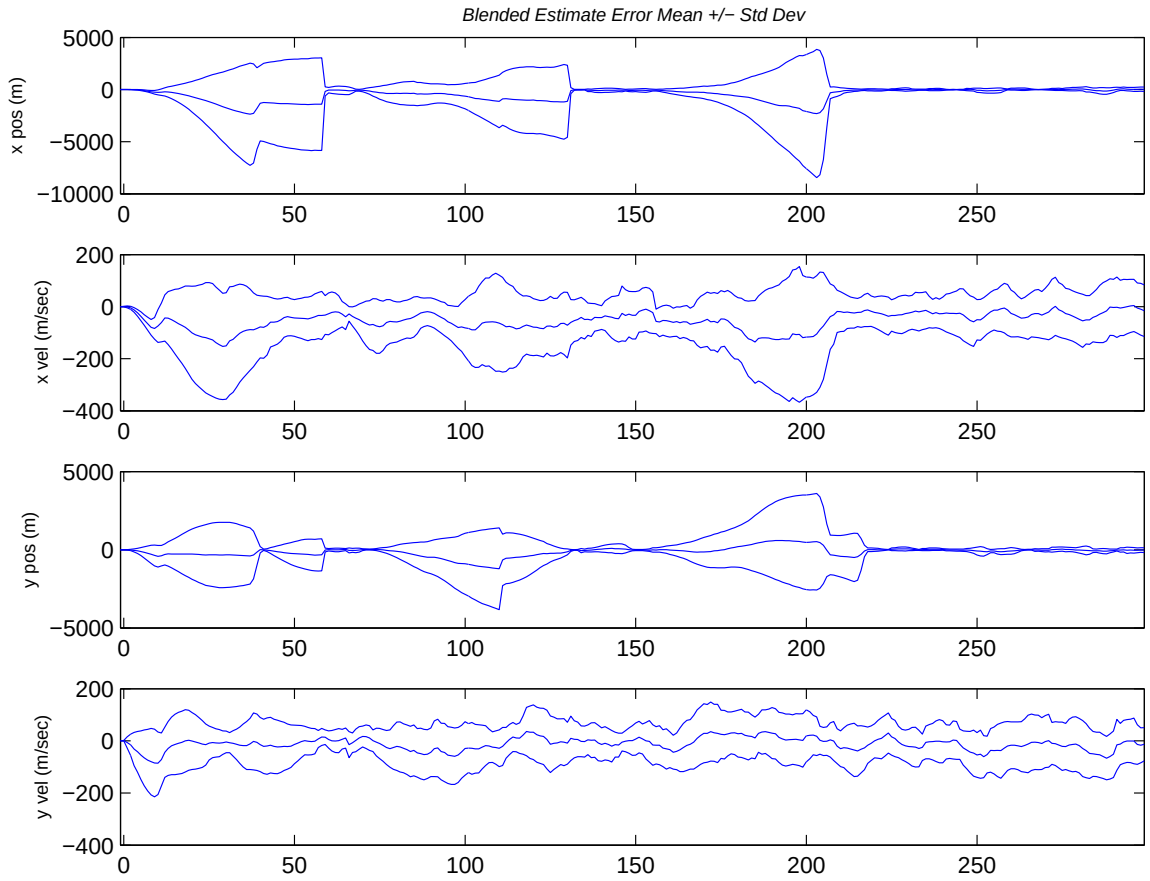

Figure 4.146: Blended estimate for the medium clutter density case. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

4.6 Summary and Conclusions on MHT Performance at all Clutter Densities

4.6.1 Clutter Performance versus Clutter-Free Performance. During the analysis of the MHT simulation results, a number of MHT algorithm performance attributes stood out as being vastly different from those seen Kalman-filter-based algorithms. It also became obvious that the MHT trackers in this research, even under the best conditions, provide state estimates that are at least one order of magnitude worse than what would be achieved using a Kalman-filter-based algorithm (with no measurement-association uncertainty, of course) with the same filter suite running against the same truth scenario without measurement clutter. In some of the worst conditions, the difference in state estimate error was several orders of magnitude worse the Kalman-based solution.

The most obvious characteristic of the MHT algorithms is their tendency to lose target "lock" under a variety of conditions, either benign clutter or heavy clutter. The best example of this was in the ultra-low clutter case in which a single FOGMA filter lost lock on a target with a time-invariant FOGMA truth model (see Fig. 4.48). The same filter suite and truth scenario in the Kalman filter case never resulted in a divergent track. In general, though, this characteristic is common to *all* tracking problems in the presence of measurement association uncertainty, and not unique to the approaches used in this research. Increasing the measurement noise covariance in the filters would force them to trust the measurements less, and outlying measurements (the ones most likely to result in track loss) would be less likely to throw the filter off course.

Unlike the Kalman-based trackers, the MHT trackers may recover from a lost lock under a variety of conditions. Theoretically, as long the true measurement appears within the measurement gate of at least one component in at least one elemental filter, there is some hope that the entire system could recover from loss of track even if all filters are doing a poor job of tracking at the present. It is possible to conceive of situations in which the true measurement has not appeared within *any* gate for a number of cycles and yet, depending on filter dynamics and clutter measurement patterns, the true measurement could reappear in a filter's gate and track would resume. This appears to happen on run number 1, filter suite 6, a two-TPV plus one FOGMA filter bank, versus scenario 6, a FOGMA truth with TPV maneuvers (see Figs. 4.86 through 4.95). Notice during sample periods ($k \approx 90, \ldots, 100$), none of the filters nor the blended estimate have a very good estimate of truth. However, after ($k = 100$), the system regains track and, once all of the TPV maneuvers have ceased, maintains track until simulation end.

Unfortunately, this tendency to lose and regain track makes performance evaluations somewhat less decisive than those that could be made for Kalman-based algorithms. For example, in Fig. 4.99, the IMM blended estimate on run number 1 is quite good and the maximum error committed never exceeds approximately 1600 meters RSS at any given time. However, on run number three, the same

IMM performed extremely poorly until the last third of the simulation. This level of discrepancy between Monte Carlo runs of the same configuration is simply not seen in the Kalman filter trials against an environment without clutter. The problem lies in the fact that one cannot entirely dismiss the runs with significant (or complete) track loss, but the runs in which track was maintained for all time give a clearer picture of what the algorithm is capable of doing in a best-case scenario.

Another significant difference in behavior between the Kalman-based and Williams-based trackers is the rapid growth of the pseudo-residual error covariance during times of track loss. In the Kalman-based algorithms, a poor matching of filter to truth will produce a very large measurement residual, but the measurement residual covariance is usually at some steady-state value once the filter has been active for awhile. The large measurement residual in the mismatched filter will normally produce a large magnitude $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ term for that sample period. However, in the MHT algorithms, a poorly matched filter seemed to produce an increasingly large computed $\mathbf{A}(k)$ measurement pseudo-residual error covariance. Presumably, this is because the mean-spreading terms in the component error covariances, $\mathbf{P}(k|k-1)$, become very large, and the component $[\mathbf{HP}(k|k-1)\mathbf{H}^{\mathrm{T}} + \mathbf{R}]$ terms become correspondingly large (ultimately leading to a large pseudo-residual error covariance). In a large number of the trials in this research, the large computed $\mathbf{A}$ would overwhelm any pseudo-residual computed by the filter, and the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ term would actually become *smaller* during times of filter/truth mismatch. Figure 4.56 is a good example. Notice how filter 1's $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ magnitude becomes smaller between $(k = 100, \ldots, 199)$ when the aggressive FOGMA maneuver (which matches filter 2) is taking place. Figure 4.57 exhibits similar behavior, and its $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ terms *grow* in magnitude during the phase of flight in which this filter matches the truth model in force. Based on experience with Kalman-based multiple-model algorithms, we would expect the exact opposite (see Fig. 4.5 for the identical filter suite and truth model scenario). The effect that rapid growth in $\mathbf{A}(k)$ has on the modal probability calculations could be significant, and this phenomenon deserves further research.

Finally, the clutter-free simulation execution times differed significantly from the simulation times of the measurement clutter cases. To reiterate, a single Monte Carlo trial involved a simulation length of 300 sample times (the declared simulation termination point). Simulations were executed on Windows-based workstations with varied amounts of physical memory and varied processor speeds. All workstations used at least a 2.4 GHz Intel® Pentium® processor, although many of the simulations were run on a 2.8 GHz Intel® Xeon® workstation. The simulation times discussed here are most representative of the performance on the Xeon® workstation.

Execution times for the Kalman-filter-based algorithms were extremely short. On average, a single-run trial took less than 1.0 second to complete. This includes overhead associated with data

file output and a modest amount of output to the "standard output" terminal window. Depending on the particular configuration, 10 Monte Carlo trials needed approximately 6 seconds of real computing time.

The MHT algorithms required substantially more time to execute the same configurations in the various measurement clutter environments. A typical ultra-low-clutter trial needed about 42 seconds of real computing time. At low measurement clutter density (expected 5 measurements/cycle), a filter suite with two FOGMA models or two TPV models and one FOGMA model took approximately 4 hours to complete a single trial on the Xeon® workstation. The peak memory (RAM) requirement during these trials was between 100-200 megabytes. The medium-clutter trials (expected 10 measurements/cycle) required an approximate 50% increase in execution time and exhibited a peak RAM usage of about 300-400 megabytes. Execution times and peak memory requirements are greatly affected by the maximum number of components existing within a filter's Gaussian mixture after a Williams reduction cycle, and this number can be specified at simulation start (see the introduction to Chapter 4, Section 4.5 for a more detailed treatment of this trade-off). Nonetheless, the MHT algorithms, as implemented here, are far too slow for real-time application. Chapter 5 will discuss proposed enhancements that could significantly reduce the computational requirements of the Williams-filter-based algorithms.

4.6.2 MMAE versus IMM Performance. Like the no-clutter cases, making an assessment about the relative performance of MMAE and IMM proves to be rather difficult. In the low-clutter cases, MMAE clearly outperformed IMM on filter Suite 4 versus truth model Scenario 4, which includes the piecewise-time-varying FOGMA maneuver. IMM did slightly better than MMAE on Suite 5 versus Scenario 5, which includes three, +3-g TPV maneuvers. MMAE outperformed IMM again on Suite 6 versus Scenario 6, where the underlying benign truth model is a FOGMA in place of Scenario 5's constant-velocity model.

In the medium-clutter environment, IMM did extremely well with Suite 4 versus Scenario 4, even though it did worse than MMAE for the same case in the low-clutter environment. MMAE performed on-par with its performance in the low-clutter case. Due to time limitations, no IMM case was run for Suite 5 versus Scenario 5 at the medium-clutter case. MMAE performed about as well here as it did for the same Suite/Scenario matching in the low-clutter case. IMM did slightly better than MMAE when Suite 6 was run against Scenario 6, although differences in plot scaling obscure the similarities in performance.

The medium-clutter case of the IMM with Suite 4 running against Scenario 4 represents, by far, the best performance blended estimate performance of any of the MHT trials. This author believes *both* MMAE and IMM could consistently produce similar results if the measurement gating issues

are resolved. This particular case may represent a particularly good measurement gate environment, because the results seem much closer to the Kalman-filter-based algorithm performance than they do to any of the other MHT trials.

In addition to the measurement gate behavior, concerns about the way in which ever-increasing filter-computer pseudo-residual error covariances dominate the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ computation make this author reluctant to declare MMAE or IMM better than the other. If both of these issues were to be resolved, both algorithms would hopefully perform like the IMM discussed above. Furthermore, the ideal test for deciding between the two algorithms would involve truth model dynamic which: 1) do not exactly match the dynamics of any of the elemental filters, 2) involve continuous or nearly continuous variation of dynamics parameters rather than the piecewise-time-varying models used here, and 3) are generated using realistic flight trajectories generated from high-order models rather than assumed flight trajectories generated from models of the same order as the elemental filters themselves. It is felt that these three enhancements would better replicate a real-world tracking environment and would better distinguish the performance of MMAE and IMM.

V. Conclusions and Recommendation

5.1 Restatement of Research Goals

As stated in Section 1.1, the goal of this research has been to develop a Multiple Hypothesis Tracker (MHT) using multiple, Williams Integral Square Error (ISE) Gaussian mixture reduction filters operating within a multiple-model, adaptive estimator architecture. The multiple-model architecture addresses the different hypothesized models for best describing a range of possible target trajectory variations, whereas the Williams filter (versus the conventional Kalman filter) form for elemental filters addresses the presence of measurement clutter. Previous research by Williams [30] showed the ISE-based mixture reduction algorithm performed better than alternative mixture reduction techniques, and the summary of results were outlined in Chapter 2, Section 2.3.6. Conventional multiple-model estimator algorithms, namely the Multiple Model Adaptive Estimator (MMAE) and the Interacting Mixed Model (IMM) estimator, were adapted to perform as MHT's by removing their standard Kalman filters and replacing them with Williams filters. Chapter 3, Section 3.3 discussed, in detail, the steps necessary to create Williams-based elemental filters which produce equivalent outputs to those produced by an elemental Kalman filter in a conventional MMAE or IMM algorithm.

Chapter 4 demonstrated that MMAE and IMM implementations using conventional Kalman filters functioned as expected. Using various elemental filter suites running against piecewise-time-varying (maneuvering) truth models in scenarios without measurement association uncertainty (i.e., no clutter measurements), the performance of these multiple-model algorithms was evaluated. The performance exhibited by these configurations was considered broadly representative of the *maximum possible* performance achievable by identically equipped MHT, Williams-filter-based configurations operating in measurement clutter environments.

Using the Williams-filter-based algorithms, the same filter suites and truth model scenarios were run in "ultra-low"-clutter, low-clutter, and medium-clutter environments. Insight gained from the analysis of the clutter-free cases aided the analysis of the clutter cases. Performance comparisons were made between similar configurations run at the various levels of clutter, and conclusions were drawn about the fundamental behvior differences between the Kalman-filter-based algorithms and the MHT, Williams-filter-based algorithms. In addition, the relative performance of MMAE versus IMM was evaluated.

5.2 Summary of Results

1. The multiple-model algorithms performed as expected in the clutter-free environment. In general, filter suites that were well-tuned to the assumed truth model scenarios exhibited

performance that was far superior to that achievable by a single, time-invariant-dynamics-model Kalman filter running against the same set of maneuvers. Furthermore, during the piecewise-time-invariant phases of flight, the elemental filters performed exactly as a single, time-invariant-dynamics-model Kalman filter would if it matched the assumed truth model in force for all time. State estimation error was essentially zero-mean under all scenarios that were piecewise-matched (i.e., one elemental filter in the suite matched the truth dynamics in force for all phases of flight for all time, see Section 4.1.4). The actual error committed by the tracker blended estimate, on average, had a mean $\pm 1\sigma$ envelope that stayed within approximately 1 to 2 meters of zero in position for all truth dynamics phases, and 3 to 5 meters/sec in velocity for benign truth dynamics phases. During aggressive maneuver phases, the mean $\pm 1\sigma$ envelope increased to staying within zero $\pm$ 8 to 10 meters/sec, on average.

2. In cases involving measurement clutter, the MHT algorithms using Williams ISE elemental filters performed well enough to give confidence in both the theoretical foundation used in their development as well as the specific techniques used in the software implementation. For the majority of test scenarios, position and velocity state estimate error standard deviations were about three orders of magnitude larger than those in the identically configured Kalman-filter-based algorithms running in a clutter-free environment.

From the perspective of algorithm verification, the IMM with filter Suite 4 running against truth model Scenario 4 (see Appendix B for descriptions of the suites and scenarios) in the medium-clutter environment (expected 10 measurements/cycle) produced the most significant results in this research. Position error mean $\pm 1\sigma$ was within approximately 5 meters and velocity error mean $\pm 1\sigma$ was within approximately 20 meters/sec during the aggressive FOGMA maneuver and approximately 10 meters/sec during the benign phases. This performance was, in fact, significantly better than the same configuration's performance in the low-clutter environment (expected 5 measurements/cycle).

The conclusion is that this specific configuration in the medium-clutter environment benefitted from particularly good measurement gating situations at each sample period. Overall algorithm performance was much closer to the performance anticipated by the theoretical developments in Chapters 2 and 3 than were any of the other MHT configurations tested. The equivalent MMAE (filter Suite 4 versus truth model Scenario 4) performed about the same in all clutter cases as the IMM did in the low-clutter case. Furthermore, ignoring the special case IMM's with only a single elemental filter, the overall relative performance of MMAE versus IMM was difficult to characterize as *better* or *worse*, because both algorithms performed well in some configurations and worse in others. With that in mind, this author believes the IMM

running Suite 4 against Scenario 4 in the medium-clutter case is an excellent example of what a Williams-filter-based, multiple-model MHT is capable of doing when measurement gating is functioning as expected. It is not, however, a result of any unique capability of IMM or the particular filter suite used.

3. Williams ISE-based elemental filters often exhibited much larger than anticipated filter-computed pseudo-residual error covariances at all clutter densities, even though the identically configured Kalman-filter-based algorithms operating in clutter-free environments did not exhibit this behavior. These large covariances dominated the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ computation which is so important to the multiple-model algorithm modal probability calculations. It is likely that this problem is directly or indirectly related to the way in which measurement gating takes place in the multiple-model implementations. If this issue were to be resolved, it is likely that the performance of all configurations in all clutter environments would approach that exhibited by the unusually good IMM case described above.

4. The MHT algorithms at all clutter densities created significantly more computational burden than the identically configured Kalman-filter-based algorithms operating in a clutter-free environment. This was, to a large extent, anticipated based on the knowledge of the way measurements would be processed in the elemental Williams filters. While the computational speed and memory requirements were acceptable for research-oriented simulations, these algorithms, in their current implementation, are far too slow for real-time tracking, given current computing technology. At low measurement clutter density (expected 5 measurements/cycle), a filter suite with two FOGMA models or two TPV models and one FOGMA model took approximately 4 hours to complete a single, 300-sample-time Monte Carlo trial on a 2.8 GHz Intel® Xeon® workstation. The peak memory (RAM) requirement during these trials was between 100-200 megabytes at the low clutter density and about 300-400 megabytes at the medium clutter density. A corresponding trial for the Kalman-filter-based algorithm (by definition, operating without the presence of measurement clutter) took less than 1.0 second, on average, depending on the configuration.

5.3 Significant Contributions of Research

1. The concept of a multiple-model adaptive estimator containing Williams ISE-based Gaussian mixture reduction filters, in place of conventional Kalman filters, is a conceptually sound and computationally realizable approach to Multiple Hypothesis Tracker design. This research succesfully implemented a fixed-model type Multiple-Model Adaptive Estimator (MMAE) and

switching-model type Interacting Multiple Model (IMM) tracker using elemental Williams filters.

2. Using conventional Kalman-filter-based algorithms, various filter suites were tested against various piecewise-time-varying truth models to gain insight into filter combinations that would make worthwhile MHT configurations. Performance of the Kalman-filter-based algorithms was consistent with the known capabilities of these trackers operating in a clutter-free measurement environment.

3. The MHT algorithms, using the same filter suites analyzed in the clutter-free environment, were tested in three different measurement clutter density environments against the same piecewise-time-varying truth models used in the clutter-free cases. Their performance in these environments was sufficiently good to verify the validity of the developments in Chapters 2 and 3. Furthermore, algorithm behavior during Monte Carlo trials provided considerable insight into potential software enhancements which may produce dramatic improvements in tracker performance at all clutter densities.

4. Partition-sensitive probabilities were successfully implemented using the block system technique described in Chapter 3, Section 3.2. This allowed elemental filters of different state-dimensionality to run alongside each other in the same bank of filters belonging to a particular IMM or MMAE configuration. Furthermore, this technique preserved *all* of the filter state estimate and error covariance information available within in a particular bank of mixed-dimensionality filters. That is to say, tracker blended estimates were the same dimension as the highest dimensioned filter in the bank, even if some of the states were not present in all of the elemental filters. In such cases, the information in the blended estimate represented partition-sensitive probability-weighted summations of the actual estimates provided by all of the filters and not some *ad hoc* representation of those states. This allowed the MMAE algorithms to provide *full-dimensioned* estimates and error covariances to any filter in the system during a divergent filter reset[1]. This technique also allowed IMM mixing cycles to provide *full-dimensioned* estimates and covariances to any filter in the system.

   Truth models *could* possess a state-dimensionality that was different from any or all of the filters in the tracker, but these cases do not benefit from the block system representation. Truth models *could not* change dimensionality within the same simulation (i.e., a 4-state truth

[1] Full-dimensioned implies that the filter undergoing the reset (MMAE) or receiving the mixed estimate (IMM) receives a state estimate and error covariance equal to its full dimension. All states in that estimate and all terms in that covariance are representative of actual probability-weighted estimates and covariances available within the tracker (as opposed to some equivalent in which some states and covariance terms are explicitly set in an *ad hoc* manner)

model could not change to a 6-state model at any point in the simulation), but truth models of arbitrary dimension may be configured prior to simulation start.

5.4 Recommendations for Future Research

1. By far, the most fruitful improvement to the Williams-filter-based MHT algorithms would be a change in the measurement gating approach. The unioning of measurement gates was a necessary modification, but it was chosen mostly for its simplicity of implementation. On one hand, unioning acceptably accomplishes the primary goal of measurement gating: eliminate the need for the tracker to process all measurements within the measurement space. On the other hand, unioning causes any given elemental filter to maintain association histories involving potentially many measurements that are *only* part of the unioned set and *not necessarily* within the gate of the particular filter of interest.

   The conclusion drawn from the simulation results is that unioning of gates caused the especially pessimistic pseudo-residual error covariances seen in the Williams-filter-based algorithm trials. The component residuals (existing in a particular filter's pseudo-residual Gaussian mixture) that are *only* part of the unioned set for the particular filter will, necessarily, be very large compared to the residuals which resulted from measurements gated by the filter itself. In all likelihood, the residuals gated by the filter itself will have the bulk of the probability weight and, consequently, will most heavily influence the mean of the resulting pseudo-residual. Looking at Eq. (3.8), the larger unioned residuals will, however, create significant mean-spreading terms and drive up the eigenvalues of the pseudo-residual error covariance. This explains the apparent contradiction between the *actual* pseudo-residual error covariance (as computed from the Monte Carlo run data sets) and the *filter-computed* pseudo-residual error covariance (as computed within the elemental filters themselves) in the Williams-filter-based trials.

   There are several alternative approaches which, like unioning, would ensure all elemental filters receive the same sets of measurements. One option is to create a single, very large rectangular gate with an area that encompasses every component estimate in every elemental filter. As the rectangle becomes larger, it more closely approximates a system with no measurement gating at all (i.e., a system that processes all measurements in the space). This would be simple to implement, but it would most likely require many more measurements to be processed than truly necessary.

   Another approach is to create a *convex hull* that encompasses the gates of all elemental filters with a convex polygon possessing a variable number of sides and covering a variable

amount of the total measurement space. This technique strikes a balance between the computationally simple (yet functionally insufficient) unioning technique with the rectangular gate technique that causes high computational intensity due to so many (functionally unnecessary) measurements being admitted. Forming a convex hull can, itself, be somewhat computationally challenging, but it is probably much less challenging than the processing burden and memory storage burden of processing all of the measurements gated by a conservatively large rectangular gate.

Late in the research process, an idea was proposed that in some ways challenges the concept of the rectangle and convex hull approach. Compared to the convex hull and large rectangular gates, unioning offers a significant *reduction* in the number of total measurements processed. However, compared to no unioning whatsoever (i.e., each filter only processes the measurements within its *own* component gate), filter gate unioning represents a significant *increase* in computational demands due to the higher number of measurement/component associations required at each cycle. The new proposal favored the unioning approach, but instead of every filter processing every measurement in a unioned set, any given filter would process all measurements within its own gate and *one* measurement from the gate of every other elemental filter in the system. Ideally, the measurement from the other filters would be the measurement that created the component residual with the highest probability weight *within the filter in which it was originally gated.*

Compared to the original gating routine, this significantly cuts down on the number of association hypotheses generated at each update cycle. Furthermore, within any given elemental filter, the inclusion of the single measurements from every other filter will create a small number of association hypotheses that force the Williams filter to evaluate the possibility that its state estimate is, indeed, rather far away from truth. In other words, the modified union forces the acknowledgement of a potentially far away true measurement (which is precisely what drives the probability flow in a multiple-model algorithm). At the same time, the single measurement from each filter would reduce the influence the unioned set has on the mean-spreading term in the pseudo-residual error covariance computation. That too would aid probability flow by reducing the dominance of the filter-computed $\mathbf{A}(k)$ term in the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ computation.

There are two foreseen downsides to this approach, but their practical effect is virtually impossible to predict without further research. The first is that unioning would still ignore some portions of the measurement space existing between filter estimates (a space in which the true measurement is moderately likely to lie). Depending on the particular clutter environment

and truth trajectory situation, the true measurement could exist in this ignored space for a considerable amount of time, yet no filters would ever gate the true measurement. An MHT's deferred decision-making would be seriously hurt by such a scenario.

The second downside is that each filter would be receiving different measurement histories for formation of the pseudo-residuals. Remember that all of the multiple-model algorithm derivations (and the underlying Kalman filter derivations) include probability density functions conditioned on the measurement history $\boldsymbol{Z}^k$, and in the MHT case, also conditioned upon particular measurement association histories $\boldsymbol{\Psi}_u(k)$ for all $u = 1, \ldots, N_h(k)$. In the Kalman-filter-based multiple-model algorithms operating without measurement clutter, the conditioning upon $\boldsymbol{Z}^k$ implies that all filters receive the *same* measurement vector $\boldsymbol{z}(k)$ during the residual formation and update cycle. Likewise, the original union concept ensured that each filter received the same *augmented* measurement vector $\boldsymbol{Z}_k$ during the residual formation and update cycle. However, the modified union concept deviates somewhat from this idea, because, for any given elemental filter, only a single measurement from each of the alternative elemental filters is passed to the unioned set. In other words, the augmented measurement vector $\boldsymbol{Z}_k$ is different for each elemental filter.

Such an argument may be mostly academic. In practice, the small probability weights assigned to the associations created by unioned measurements in the original concept will have little effect on the pseudo-residual Gaussian mixture. Removing these measurements from the set might improve performance considerably by reducing computation loads yet maintaining the possibility (within each filter) that other filters offer better estimates of the truth state. Ultimately, the effects caused by the different augmented measurement vectors and the measurement space gaps would best be evaluated in further research efforts.

2. A significant streamlining of the C++ code should be undertaken. Use of symmetry in the Gaussian component covariance terms would greatly improve the speed of computation as well as significantly reduce the total memory storage requirements. Furthermore, the code was implemented to be as similar as possible to the conceptual algorithm implementations described in Chapters 2 and 3. Now that the code operation has been verified to the maximimum extent possible, many of the most computationally intensive portions of the code could undergo substantial reductions in the number of matrix operations, memory allocations, and function calls. Such enhancements would hopefully drive the tracker execution speed towards something closer to real-time (although that would require orders-of-magnitude improvement in speed for the low and medium-clutter cases) and allow practical testing of configurations

using more than three elemental filters (which might operate more slowly than real-time but faster than such configurations would operate in the current implementation).

3. Some provision should be made for simulations using more realistic truth trajectories, especially trajectories produced by a specialized trajectory generation routine. Such trajectories would, ideally, be based on higher-order dynamics models than those modelled in the filter banks. These trajectories should also be based on continuously varying dynamics parameters rather than the piecewise-constant parameters used in this research. Ultimately, the goal would be to use trajectories that represent realistic airborne vehicles operating in a variety of flight regimes performing maneuvers ranging from benign to very aggressive. Performance comparisons between MMAE and IMM, and among different filter suites, would be much more insightful because the simulated targets of interest, like real-world targets, would exhibit a blending of dynamics which, by definition, could not be represented by any single, statically declared, time-invariant-dynamics-model-based elemental filter.

Appendix A. Plot Explanations

This appendix discusses the attributes of the various plot styles used in this research. Note that the axes scalings may differ between two plots of the same type, because the scalings are designed to maximize readability and proper data fit.

A.1 Elemental Filter Summary Plots

These plots show statistical data computed from a collection of Monte Carlo runs. Except where explicitly noted, these plots are computed using 10 Monte Carlo runs. Also, except where noted, the 10 Monte Carlo run random seed numbers are consistent across plots (i.e., the same 10 random seed numbers drove all simulations in this research). This allows direct comparisons between trackers of different configuration and on a sample-by-sample basis. All plots show the number of sample periods along their horizontal axis. Note that the error plots (leftmost column, rows 1 through 4) are defined to be components of $[\hat{\boldsymbol{x}}(k|k) - \boldsymbol{x}_{truth}(k)]$.

The leftmost column of every figure pertains to actual estimation performance, whereas the rightmost column pertains to residuals and computed probabilities associated with the adaptation within the multiple-model algorithm. Filter-computed values are always plotted as wide gray lines while sampled data is plotted as thin, solid black lines. Starting with column one of this plot and moving down the columns:

1. X-position mean error (in meters), with true mean $\pm 1\sigma$ (computed from data) overlayed on top of $\pm 1\sigma$ (filter-computed)

2. X-velocity mean error (in m/sec), with true mean $\pm 1\sigma$ (computed from data) overlayed on top of $\pm 1\sigma$ (filter-computed)

3. Y-position mean error (in meters), with true mean $\pm 1\sigma$ (computed from data) overlayed on top of $\pm 1\sigma$ (filter-computed)

4. Y-velocity mean error (in m/sec),with true mean $\pm 1\sigma$ (computed from data) overlayed on top of $\pm 1\sigma$ (filter-computed)

5. X-axis measurement residual (in meters), with true mean $\pm 1\sigma$ (computed from data) overlayed on top of $\pm 1\sigma$ (filter-computed)

6. Y-axis measurement residual (in meters), with true mean $\pm 1\sigma$ (computed from data) overlayed on top of $\pm 1\sigma$ (filter-computed)

7. Filter computed $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$, with true mean $\pm 1\sigma$ directly comparable to $s = 2$ (the dimension of the measurement $\boldsymbol{z}(k)$)

8. Elemental filter modal probability (includes lower-bounding effects). In the probability plots (rightmost column, last row), the thick center line is the mean of probabilities from all Monte Carlo runs at the particular sample index. The upper envelope line is the *maximum* probability this filter took during any of the included Monte Carlo runs at the particular sample index. The lower envelope line is the *minimum* probability this filter took during any of the included Monte Carlo runs at the particular sample index. The area within the envelope is shaded for easier reading. During times in which the filter was consistently weighted the same across all runs, the envelope may be small or non-existent. Early versions of these plots plotted probabilities as mean $\pm 1\sigma$. For the Kalman-filter-based trials without measurement clutter, this approach worked well and the probability plots were easily interpreted. However, the MHT algorithms produced modal probabilities with substantially more variation across Monte Carlo trials, and the mean $\pm 1\sigma$ lines obscured filter performance sufficiently to make interpretation extraordinarily difficult. Hence, the mean $\pm 1\sigma$ probability plots were entirely discarded in favor of the mean overlayed with maximum and minimum values. For the Kalman-filter-based trials, the new plots were remarkably similar to the original plots (because those trials produced very small standard deviations), while the MHT plots became more readable and insightful.

A.2 Probability Flow Plots

This is a collection of modal probability plots for all filters in the multiple-model estimator. Individual plots are identical to the elemental filter modal probability plot described above. Performance attributes relating to probability flow between filters, the relative quality of the match between an elemental filter's model dynamics and that of the truth model, and parameter space discretization can be ascertained from direct comparison of the individual graphs of these plots. Interpretation of plots created for MMAE trackers is rather straightforward. Agile probability flow between filters during times of truth model change (maneuvers) will result in clearly-defined shifts on the affected filters. Filters that closely match the truth model dynamics will, generally speaking, have higher probabilities than other filters in the bank that do not match truth as well. Finally, a poorly chosen parameter-space discretization will often result in ambiguous probability flows during truth maneuvers and/or diluted weightings during times when the truth model is not changing.

Interpretation of plots created for IMM trackers using a Markov probability transition matrix other than identity is less straightforward. Non-identity Markov matrices cause state estimate interaction among all elemental filters according to the Markov matrix off-diagonal terms. Therefore, at every sample period, the state estimate in *any single elemental filter* is the product of *a history of dynamics propagations* according to a Markovian sequence defined by the values in the

Markov matrix itself. Contrast this to MMAE, in which an elemental filter's estimate at every sample period is the product of a dynamics propagation using *only* the dynamics assumed for that particular elemental filter (assuming divergent filter restarts have not taken place for some time). The IMM trackers will exhibit more ambiguous probability flows because each elemental filter does not precisely correspond to a single particular *truth* dynamics model for all time. Rather, each filter represents a probabilistically weighted hybridization of all dynamics models in the system.

With these differences in mind, an assessment of IMM filter bank performance *can* be made with the understanding that filters closely matching truth dynamics will usually have higher probability weights than filters which poorly match truth. Compared to MMAE, however, the relative weightings will be more uniformly distributed across all filters in the bank.

A.3 Single Monte Carlo Run Plots

While data collected from a single Monte Carlo run should not be considered statistically meaningful, it can offer several insights into algorithm performance that cannot be ascertained from the statistical plots described above. Truth trajectory plots give insight into the relative aggressiveness of a maneuver, and they will be included when appropriate. Single-run plots showing blended tracker output can show how well the tracker adapted to a maneuver, and it is helpful in determining when the tracker output lost lock or diverged completely from truth. Oftentimes, variations in track loss behavior between different tracker configurations will be more visible in one of these plots than it will be in the statistical computation plots. Where appropriate, error plots showing truth position minus filter-computed position will be provided.

The single-run plots used in this document include three panes. The top pane is the filter (or tracker blended) position estimate overlayed on top of the truth trajectory. The filter (or blended) estimate is denoted by black circle data markers and the truth is denoted by gray "star" data markers. A large triangle marks the initial truth position and a large square denotes the final truth position allowing the reader to follow the target's motion from sample ($k = 0$) through simulation termination. The second and third panes are x-position and y-position error (filter-computed value minus truth or blended output minus truth) versus sample period.

Appendix B. Filter Model Suites, Truth Trajectory Scenarios, and IMM Markov Matrices

Note that all of these filter suites were tested, but some of them produced results that were of little interest because they closely resembled the performance of other suites and scenarios. Those simulations of little interest may not be reflected in the analysis of Chapter 4.

B.1 Continuous Time Dynamics Representations

Constant-Velocity (CV) Models

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ y(t) \\ v_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{B.1}$$

where acceleration is modelled by additive zero-mean, Gaussian, white-noise with mean and covariance kernel:

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\,\delta(t-t') \end{aligned} \tag{B.2}$$

where $\mathbf{Q} = q\mathbf{I}$, and q is the strength of the Gaussian white noise [18].

Thrust-Perpendicular-to-Velocity (TPV) Models

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ y(t) \\ v_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_x & 0 \\ 0 & 0 \\ 0 & b_y \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{B.3}$$

where the b terms are recalculated at each sample instant based on the current vehicle velocity magnitude and direction. Specifically, the b terms obey the relationships (in terms of speed s and the acceleration of gravity equal to 9.8 m/sec^2):

$$b_x = -v_y/s \cdot \mathrm{GF} \cdot 9.8 \tag{B.4}$$

$$b_y = v_x/s \cdot \mathrm{GF} \cdot 9.8 \tag{B.5}$$

$$s = \sqrt{v_x^2 + v_y^2} \tag{B.6}$$

where GF represents a scale factor on the TPV g-loading, such that GF = 1.0 implies the vehicle is undergoing a 1.0-g turn. In addition to the deterministic control inputs, acceleration is modelled by

additive zero-mean, Gaussian, white-noise with mean and covariance kernel

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\,\delta(t-t') \end{aligned} \tag{B.7}$$

where $\mathbf{Q} = q\mathbf{I}$, and q is the strength of the Gaussian white noise [18].

First-Order Gauss Markov Acceleration (FOGMA) Models

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{a}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \\ \dot{a}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ a_x(t) \\ y(t) \\ v_y(t) \\ a_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{T} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{T} \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{B.8}$$

where acceleration is a first-order (exponentially time-correlated) Gauss-Markov process, the output of a first-order lag (with lag coefficient $\frac{1}{T}$) driven by zero-mean, white, Gaussian noise with mean and covariance kernel

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\,\delta(t-t') \end{aligned} \tag{B.9}$$

where $\mathbf{Q} = q\mathbf{I}$, and

$$q = 2 \cdot T \cdot \sigma_a^2 = 2 \cdot T \cdot \mathrm{QF} \cdot (9.8)^2$$

The QF term is a scale factor based on the g-loading, and the $1/T$ term in the $\mathbf{G}(t)$ matrix permits independent tuning of T and σ_a^2 (the rms acceleration) without altering the gain associated with the low-frequency asymptote on a FOGMA's PSD curve. The rms g-loading is expressed as $\text{g-load}_{rms} = \sqrt{\mathrm{QF}}$. Therefore, if QF = 4.0, the rms acceleration of the target is 2.0-g or 19.6 m/sec^2.

Suite 1

Filter	Configuration
1	Constant-Velocity (CV) dynamics with $q = 1.0$

Remarks:

A constant-velocity model with low strength white noise at the acceleration level. Constant-velocity models are commonly used to model aircraft in air traffic control tracking systems. This filter matches that used by Williams [30] in his research.

Suite 2

Filter	Configuration
1	First-Order Gauss-Markov (FOGMA) dynamics with QF = 1.0, $T = 4.0$

Remarks:

A benign FOGMA filter with exponentially time-correlated acceleration. This was considered broadly representative of an aircraft operating in an environment without any known threats. The target's rms acceleration magnitude would be about 1-g with a time-constant of approximately 4 seconds.

Suite 3

Filter	Configuration
1	CV dynamics with $q = 1.0$
2	FOGMA dynamics with QF = 1.0, $T = 4.0$

Remarks:

The constant-velocity filter of Suite 1 alongside the FOGMA filter of Suite 2. The FOGMA filter was considered slightly more aggressive than the CV filter.

Suite 4

Filter	Configuration
1	FOGMA dynamics with QF = 1.0, $T = 4.0$
2	FOGMA dynamics with QF = 2.0, $T = 0.5$

Remarks:

The benign FOGMA filter of Suite 2 alongside a more aggressive FOGMA filter. The aggressive FOGMA broadly represents a more aggressive aircraft operating in a moderate threat environment. The rms acceleration magnitude is about 1.4-g's with a time constant of 0.5 seconds. The choice of q and T in the aggressive FOGMA was partially motivated by a desire for a moderately broad parameter space discretization (which accentuates probability flows in the multiple-model algorithms).

Suite 5

Filter	Configuration
1	TPV with $q = 1.0$ and g-loading of +3 g's
2	TPV with $q = 1.0$ and g-loading of -3 g's
3	CV dynamics with $q = 1.0$

Remarks:

The CV filter of Suite 1 with the addition of two TPV filters with mirrored g-loadings of ±3-g's. This configuration is tuned for target models exhibiting low strength random walk at the velocity level, even during TPV maneuvers. The target will presumably execute TPV jinking maneuvers in the presence of known and emminent threats.

Suite 6

Filter	Configuration
1	TPV with $q = 1.0$ and g-loading of +3 g's
2	TPV with $q = 1.0$ and g-loading of -3 g's
3	FOGMA dynamics with QF = 1.0, $T = 0.5$

Remarks:

The FOGMA filter of Suite 2 with the addition of two TPV filters with mirrored g-loadings of ±3-g's. This configuration is probably representative of a military aircraft's behavior, since the FOGMA model handles flight during periods of few or no threats and the TPV filters handle flight during periods of high threat. The target will presumably execute TPV jinking maneuvers in the presence of known and emminent threats.

Suite 7

Filter	Configuration
1	FOGMA dynamics with QF = 1.0, $T = 4.0$
2	FOGMA dynamics with QF = 4.0, $T = 0.5$

Remarks:

The benign FOGMA of Suite 2 is alongside a more aggressive FOGMA filter. The second FOGMA broadly represents an airborne vehicle executing aggressive maneuvers in which the rms acceleration is about 2-g's and the time constant is 0.5 seconds.

B.3 Truth Model Scenarios

Scenario 1

Phase	Description
1	Constant-velocity truth model with $q = 1.0$ for all time.

Remarks:

A target truth model that matches the dynamics of filter Suite 1.

Scenario 2

Phase	Description
1	FOGMA truth model with QF = 1.0 and $T = 4.0$ for all time

Remarks:

A target truth model that matches the dynamics of filter Suite 2.

Scenario 3

Phase	Description
1	FOGMA truth with QF = 0.5 and $T = 12.0$ for $k = 0, \ldots, 99$
2	FOGMA truth with QF = 2.0 and $T = 0.5$ for $k = 100, \ldots, 199$
3	FOGMA truth with QF = 0.5 and $T = 12.0$ for $k = 200, \ldots, 300$

Remarks:

This scenario was designed to approximate the dynamics models available in Filter Suite 3 because software limitations prevented a time-varying truth model from having a time-varying number of state variables (as would be required to transition from a true CV model to a FOGMA model).

Scenario 4

Phase	Description
1	FOGMA truth with QF = 1.0 and $T = 4.0$ for $k = 0, \ldots, 99$
2	FOGMA truth with QF = 2.0 and $T = 0.5$ for $k = 100, \ldots, 199$
3	FOGMA truth with QF = 1.0 and $T = 4.0$ for $k = 200, \ldots, 300$

Remarks:

A target truth model that matches the dynamics of filter Suite 4.

Scenario 5

Phase	Description
All	Underlying constant-velocity truth with $q = 1.0$ for all time
1	+3-g Deterministic TPV maneuver from $k = 40, \ldots, 60$
2	+3-g Deterministic TPV maneuver from $k = 80, \ldots, 120$
3	+3-g Deterministic TPV maneuver from $k = 150, \ldots, 200$

Remarks:

A target truth model that matches the dynamics of filter Suite 5. Note that Phases 1, 2, and 3 are not consecutive in time.

Scenario 6

Phase	Description
All	Underlying FOGMA truth with QF = 1.0 and $T = 4.0$ for all time
1	+3-g Deterministic TPV maneuver from $k = 40, \ldots, 60$
2	+3-g Deterministic TPV maneuver from $k = 80, \ldots, 120$
3	+3-g Deterministic TPV maneuver from $k = 150, \ldots, 200$

Remarks:

A target truth model that matches the elemental filters present in dynamics of filter Suite 6. Note that Phases 1, 2, and 3 are not consecutive in time. The remarks in Scenario 8 discuss a slightly better alternative (developed later in the course of this research) for matching filter Suite 6's dynamics model.

Scenario 7

Phase	Description
1	FOGMA truth with QF = 1.0 and $T = 4.0$ for $k = 0, \ldots, 99$
2	FOGMA truth with QF = 4.0 and $T = 0.5$ for $k = 100, \ldots, 199$
3	FOGMA truth with QF = 1.0 and $T = 4.0$ for $k = 200, \ldots, 300$

Remarks:

A target truth model that matches the dynamics of filter Suite 7.

Scenario 8

Phase	Description
Without TPV	Underlying FOGMA truth with QF = 1.0 and $T = 4.0$ during non-maneuver times
With TPV	FOGMA with QF = 0.25 and $T = 50.0$ during times of TPV maneuvers
1	+3-g Deterministic TPV maneuver from $k = 40, \ldots, 60$
2	+3-g Deterministic TPV maneuver from $k = 80, \ldots, 120$
3	+3-g Deterministic TPV maneuver from $k = 150, \ldots, 200$

Remarks:

This truth scenario attempts to replicate the dynamics of filter Suite 6 more closely than possible using the dynamics in truth Scenario 6. In this scenario, TPV maneuvers are executed on top of a very benign FOGMA with dynamics that closely resemble the underlying CV dynamics of the TPV filters in Suite 6. Note that Phases 1, 2, and 3 are not consecutive in time.

Scenario 9

Phase	Description
Without TPV	Underlying FOGMA truth with QF = 1.0 and $T = 4.0$ during non-maneuver times
With TPV	FOGMA with QF = 0.25 and $T = 50.0$ during times of TPV maneuvers
1	+4-g Deterministic TPV maneuver from $k = 40, \ldots, 60$
2	+6-g Deterministic TPV maneuver from $k = 80, \ldots, 120$
3	+8-g Deterministic TPV maneuver from $k = 150, \ldots, 200$

Remarks:

This is similar to Scenario 8 except the TPV maneuver g-loads exceed the g-loads in Suite 6's TPV filters, and the g-loads increase at each interval. This scenario was designed to test the flexibility of the two TPV filter suite against non-matching TPV maneuvers of higher strength. It was believed that TPV maneuvers with lower g-loadings than ±3-g would not present an aggressive enough maneuver to cause any difficulty for filter Suite 6. Note that Phases 1, 2, and 3 are not consecutive in time.

B.4 Markov Probability Transition Matrices

These matrices were used in the various simulation trials involving IMM's with Markov matrices other than Identity.

Two-Filter Non-Transition-Favoring Matrix

$$\begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}$$

Three-Filter Non-Transition-Favoring Matrix

$$\begin{bmatrix} 0.98 & 0.01 & 0.01 \\ 0.01 & 0.98 & 0.01 \\ 0.01 & 0.01 & 0.98 \end{bmatrix}$$

Two-Filter Specific-Model-Favoring Matrix

Note this assumes filter 1 is the favored model

$$\begin{bmatrix} 0.7 & 0.7 \\ 0.3 & 0.3 \end{bmatrix}$$

Three-Filter Specific-Model-Favoring Matrix

Note this assumes filter 3 is the favored model

$$\begin{bmatrix} 0.15 & 0.15 & 0.15 \\ 0.15 & 0.15 & 0.15 \\ 0.7 & 0.7 & 0.7 \end{bmatrix}$$

Appendix C. Track Loss Check Analysis

C.1 Introduction

Chapter 3 devoted some discussion to quantitative techniques for analyzing tracker performance. The analysis presented in Chapter 4 relied heavily on qualitative assessments made regarding the actual tracker error produced by the Monte Carlo trials. These assessments were most often made using the Monte Carlo trial summary plots as described in Appendix A, and these plots do provide a fairly complete picture of tracker performance. Most of these plots show actual performance (filter estimation error or filter residual) mean $\pm 1\sigma$ (computed from data) separately from the filter-computed covariances (the filter's internal assessment of its own performance). The filter-computed zero $\pm 1\sigma$ (computed by the filter) are overlayed on top of the data-computed covariances in the filter summary plots (described in detail in Appendix A).

The $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ plot is an exception, because it represents a fusion of actual simulation data (the residual or pseudo-residual $\boldsymbol{r}(k)$) with the filter-computed residual error covariance $\mathbf{A}(k)$ and presents it as one graph. Analysis of the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ plots gives a relatively concise presentation of how well the filter is *actually* performing compared to how well the filter *believes* it is performing[1]. Fundamentally, this representation allows *a single quantity* to define the performance of the tracker at any given sample period. As shown in Chapter 2, Sections 2.2.4 and 2.2.8, the multiple-model algorithm probability calculations rely heavily on this quantity.

Chapter 3, Section 3.4 motivated similar quantities for measuring tracker estimation performance as opposed to filter measurement residual performance. The data used in the computation of these track loss checks is identical to that presented in the basic filter summary plots, but it is fused together in the form of the quadratic quantities discussed in Chapter 3. The goal is to have a single quantity for performance comparison that reflects all of the data in the summary plots.

As described in Chapter 3, Sections 3.4.1 and 3.4.2, two track loss checks will receive emphasis here. First, a quantity called the *Position Predictor Metric* depicts $[\boldsymbol{e}^{\mathrm{T}}_{predicted}\ \mathbf{P}^{-1}_{predicted}\ \boldsymbol{e}_{predicted}]$ where $e_{predicted}$ is the error between the actual target position at sample period (k) and the predicted position at (k) derived from the filter-computed state estimate at $(k-1)$ propagated forward one sample period. For elemental filters or blended estimates containing only position and velocity states, the prediction is first-order (i.e., the target is assumed to have a constant velocity equal to the state vector's velocity state estimate and zero acceleration). For elemental filters or blended estimates containing position, velocity, and acceleration states, the prediction is second-order (i.e. the target is assumed to have a constant acceleration equal to the state vector's acceleration state estimate and zero jerk). The prediction is one sample period into the future, with a sample period, T, equaling

[1] For a given filter, the $[\boldsymbol{r}^{\mathrm{T}}(k)\ \mathbf{A}^{-1}(k)\ \boldsymbol{r}(k)]$ quantity should be approximately equal to the measurement dimension (always 2 in this research) if its hypothesis is valid. If the hypothesis is not valid, this quantity may be much larger.

1.0 second. The prediction's error covariance, $\mathbf{P}_{predicted}$, is computed as:

$$\mathbf{P}_{predicted} = \mathbf{H}_{pred}\,\mathbf{P}(k-1|k-1)\,\mathbf{H}_{pred}^{\mathrm{T}} \tag{C.1}$$

where $\mathbf{P}(k-1|k-1)$ is the filter (or blended) state estimate error covariance at sample period $(k-1)$ conditioned on measurements through $(k-1)$. In practice, one might also use $\mathbf{P}(k|k-1)$ since it represents the error covariance of the state estimate propagated forward in time one sample period according to the dynamics of the associated elemental filter. The matrix $\mathbf{H}_{pred}$ is simply the predictor coefficients on the state estimate, namely $[1 \;\; T \;\; 0.5T^2]$ for a three-state estimate (position, velocity, acceleration) or $[1 \;\; T]$ for a two-state estimate (position, velocity). The algorithm used to compute these quantities forms a partitioned $\mathbf{H}_{pred}$ to handle the full-dimensioned state estimates containing both x and y axes. The partioned matrix, arranged to conform to the dynamics models described in Appendix B (and assuming acceleration state information is available), would be:

$$\mathbf{H}_{pred} = \begin{bmatrix} 1 & T & 0.5T^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0.5T^2 \end{bmatrix} \tag{C.2}$$

The second track loss metric will be the more conventional state estimate error $[\boldsymbol{e}^{\mathrm{T}}\,\mathbf{P}^{-1}\,\boldsymbol{e}]$, where $\boldsymbol{e}$ is the actual error between the true target position and the filter-computed target position at sample period (k). The covariance, $\mathbf{P}$, will be the state estimate error covariance, $\mathbf{P}(k|k)$, at sample period (k) conditioned on measurements through (k). The resulting quantity will be called the *State Error Metric*.

Both statistics are plotted as the mean of the track loss quantities taken across all Monte Carlo trials at the given sample period. The quantities themselves are unitless and best used as a comparative measure as opposed to an absolute measure of performance.

Since only a limited number of trials were run at the medium-clutter density, this analysis will focus on the three configurations analyzed in Chapter 4, Section 4.5. Specifically, this will include MMAE and IMM running filter Suite 4 against truth Scenario 4, Suite 5 against truth Scenario 5, and Suite 6 against truth Scenario 6. These configurations will be analyzed for the clutter-free cases (which use the Kalman-filter-based algorithms in Chapter 4, Section 4.1), the low-clutter cases from Chapter 4, Section 4.4, and the medium-clutter cases.

Also, it should be noted that an alternative option is *not* to include the filter-computed covariance terms, as $[\boldsymbol{e}^{\mathrm{T}}\,\boldsymbol{e}]$ rather than $[\boldsymbol{e}^{\mathrm{T}}\,\mathbf{P}^{-1}\,\boldsymbol{e}]$. As discussed in Chapter 3, that would yield a true RMS error. Because the analysis in Chapter 4 was heavily dependent on the *actual* error committed by the trackers without regard to the filter-computed estimates of the error covariance, this

alternative option is not presented here. Certain scenarios might benefit from the exclusion of the covariance, especially if filter-computed error covariances appear to grow much faster than actual error covariances. In such a case, the covariance term would dominate the quadratic computation, and the results would be skewed considerably.

In this research, the pseudo-residual error covariances exhibited this problem, but the residuals and their error covariances are not of interest when evaluating track loss. However, the *filter-computed state estimate error covariances* suffered the opposite problem – they grew slower than the actual error in the MHT algorithms. In the clutter-free cases, the filter-computed covariances are representative of the actual error covariances computed from the data, dismissing the possibility that the basic filter suites are improperly tuned. Nonetheless, the computed-covariance terms will be used in the analysis here because, if nothing else, they give insight into areas needing further investigation. Furthermore, the results clearly show changes in state estimate error performance during maneuvers.

C.2 Suite 4 versus Scenario 4

The clutter-free results for MMAE and IMM appear in Figs. C.1 and C.2, respectively. Note that the left column shows the "Position Predictor Metric" $[\boldsymbol{e}^{\mathrm{T}}_{predicted}\,\mathbf{P}^{-1}_{predicted}\,\boldsymbol{e}_{predicted}]$ and the right column shows the conventional "State Error Metric" $[\boldsymbol{e}^{\mathrm{T}}\,\mathbf{P}^{-1}\,\boldsymbol{e}]$. The aggressive maneuvering phase from samples ($k = 100, \ldots, 199$) is obvious in both of these figures, and MMAE and IMM exhibit essentially identical performance. Notice the blended estimate Prediction quantities have a mean around 300 during the maneuver but only about 25 during benign phases. Careful inspection of the MMAE plots, in particular, will show that the blended estimate track loss measure is the same as that for the filter with the most probability weight (see Figs. 4.5 and 4.6 for probability weights) at the particular sample period of interest.

Results for the low-clutter cases (Figs. C.3 and C.4) are quite different. The MMAE exhibits quantities which are several orders of magnitude higher than they were for the same configuration in the clutter-free case. Notice, however, the maneuver period is evident in the middle phase of simulation, and notice how the tracker recovers after the cessation of maneuvers. Tracker recovery is indicated by the very quick reduction in track loss quantity magnitude, as seen after sample period 225. The IMM had a sufficient number of divergent estimates to skew the track loss checks completely, and performance appears to be quite poor. This plot provides very little insight other than showing that the tracker could not consistently recover from track loss in this configuration.

Results for the medium-clutter cases appear in Figs. C.5 and C.6. The MMAE quantities are somewhat larger than they were in the low-clutter case, but the maneuver period is still evident,

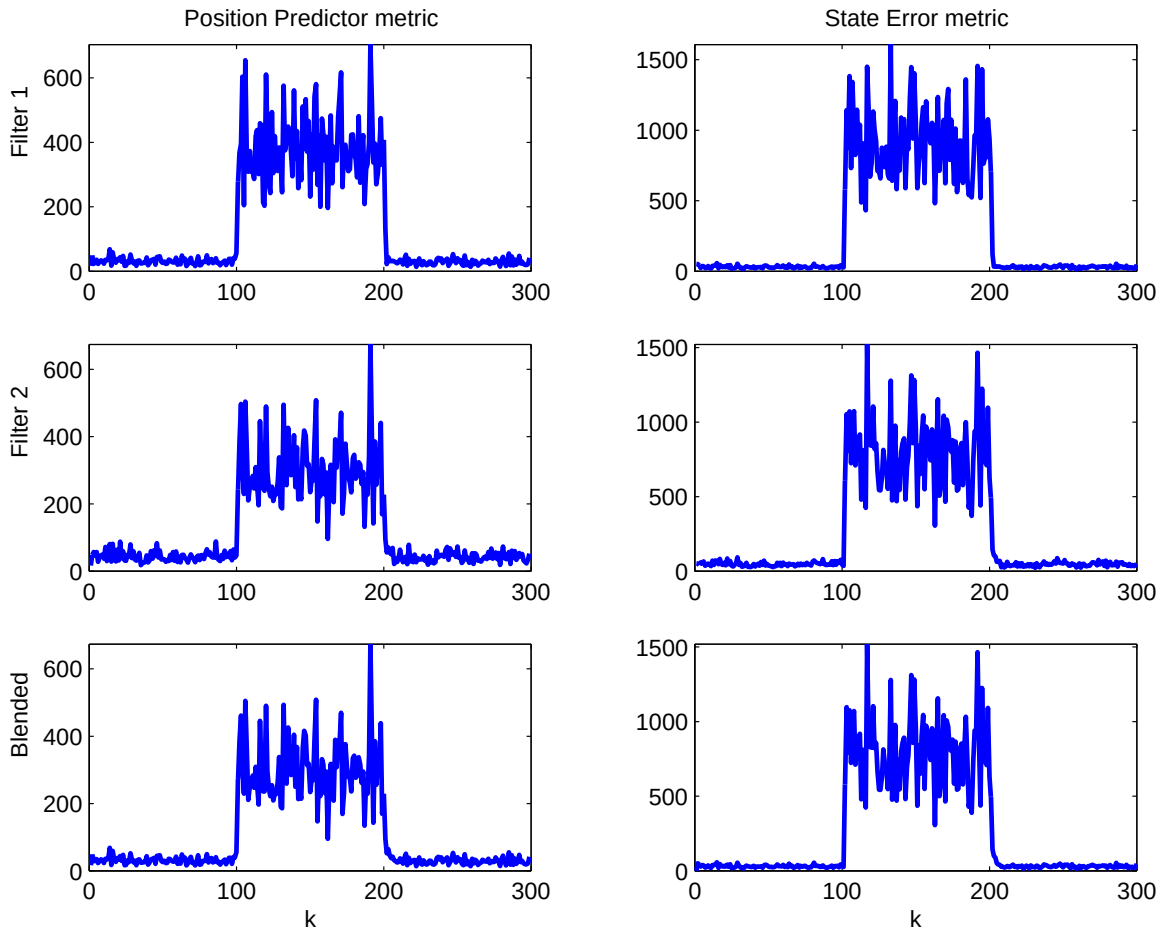

Figure C.1: Track loss results, clutter-free trials. (Suite 4 vs. Scenario 4: MMAE)

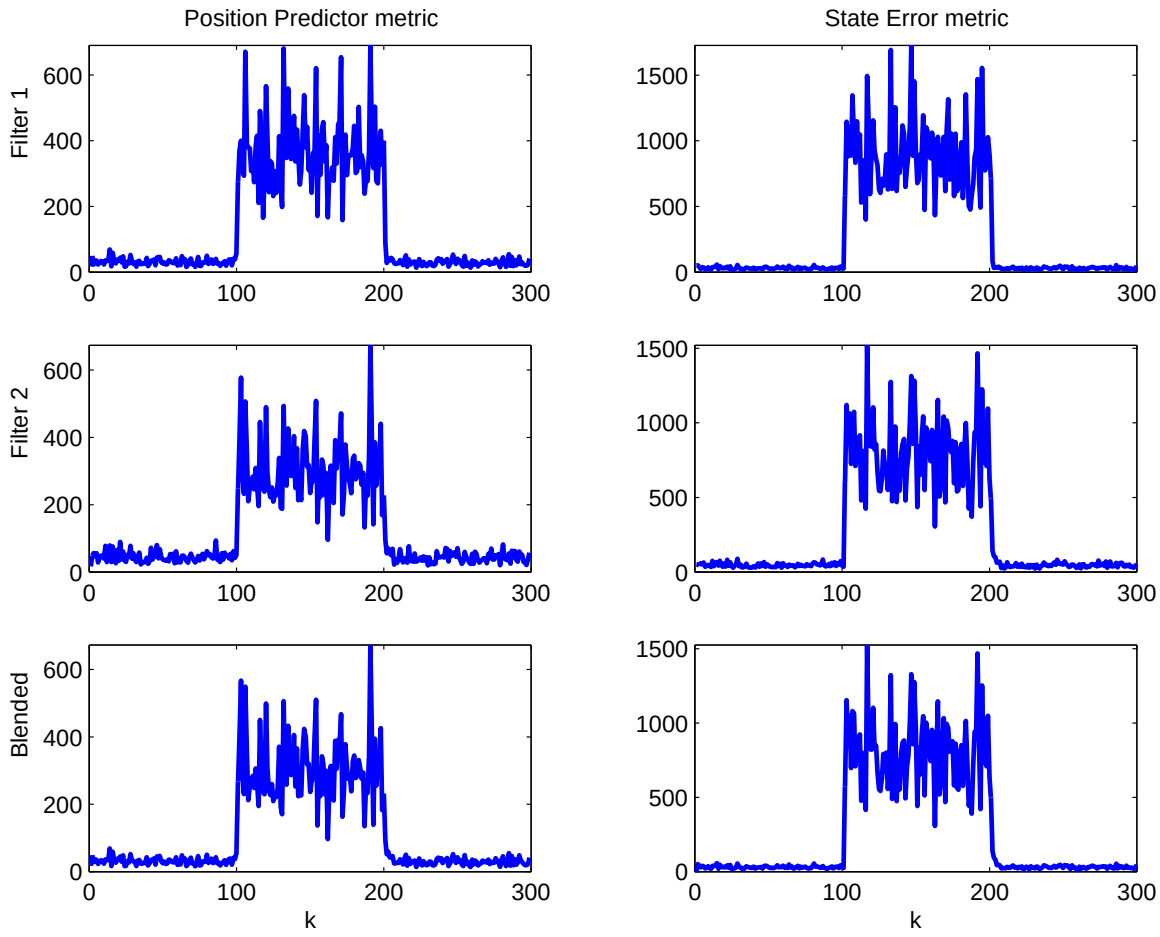

Figure C.2: Track loss results, clutter-free trials. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

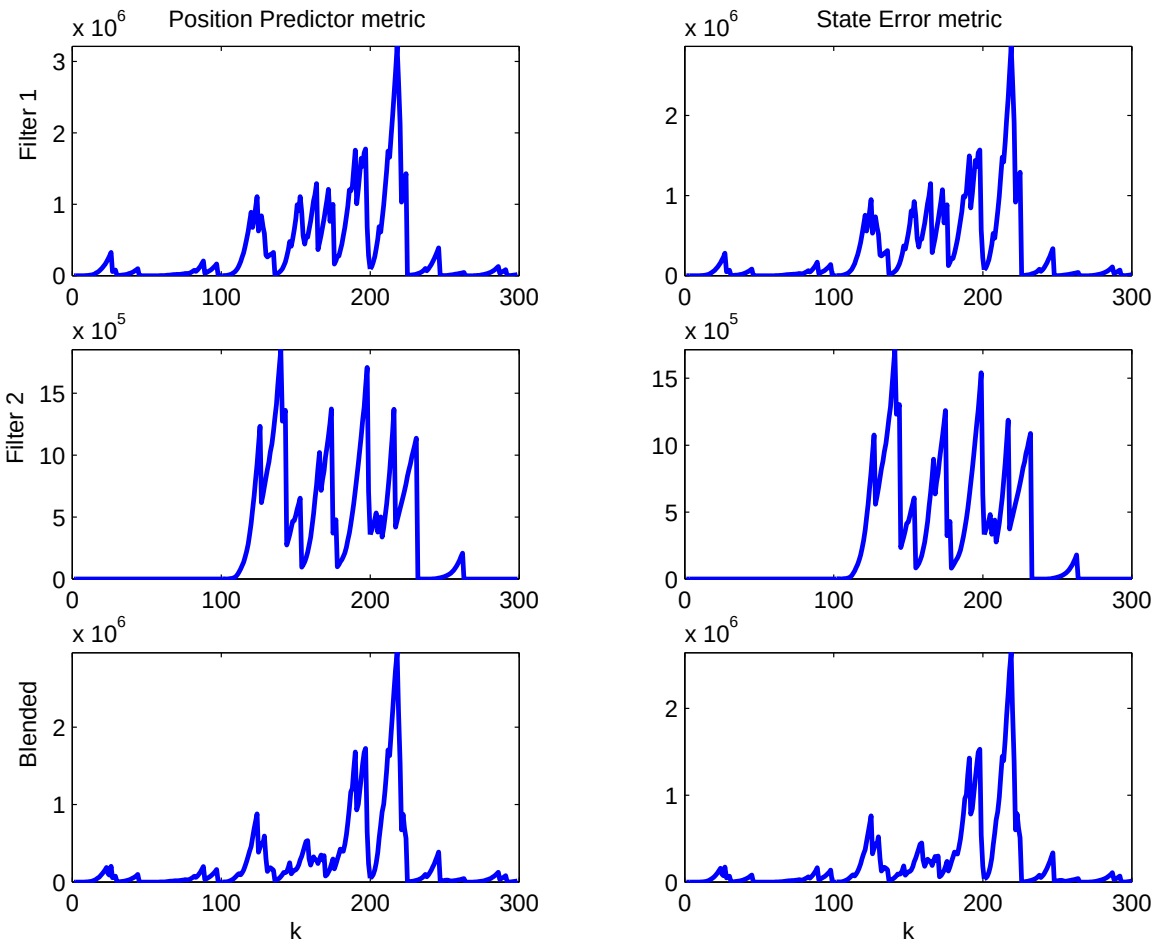

Figure C.3: Track loss results, low-clutter trials. (Suite 4 vs. Scenario 4: MMAE)

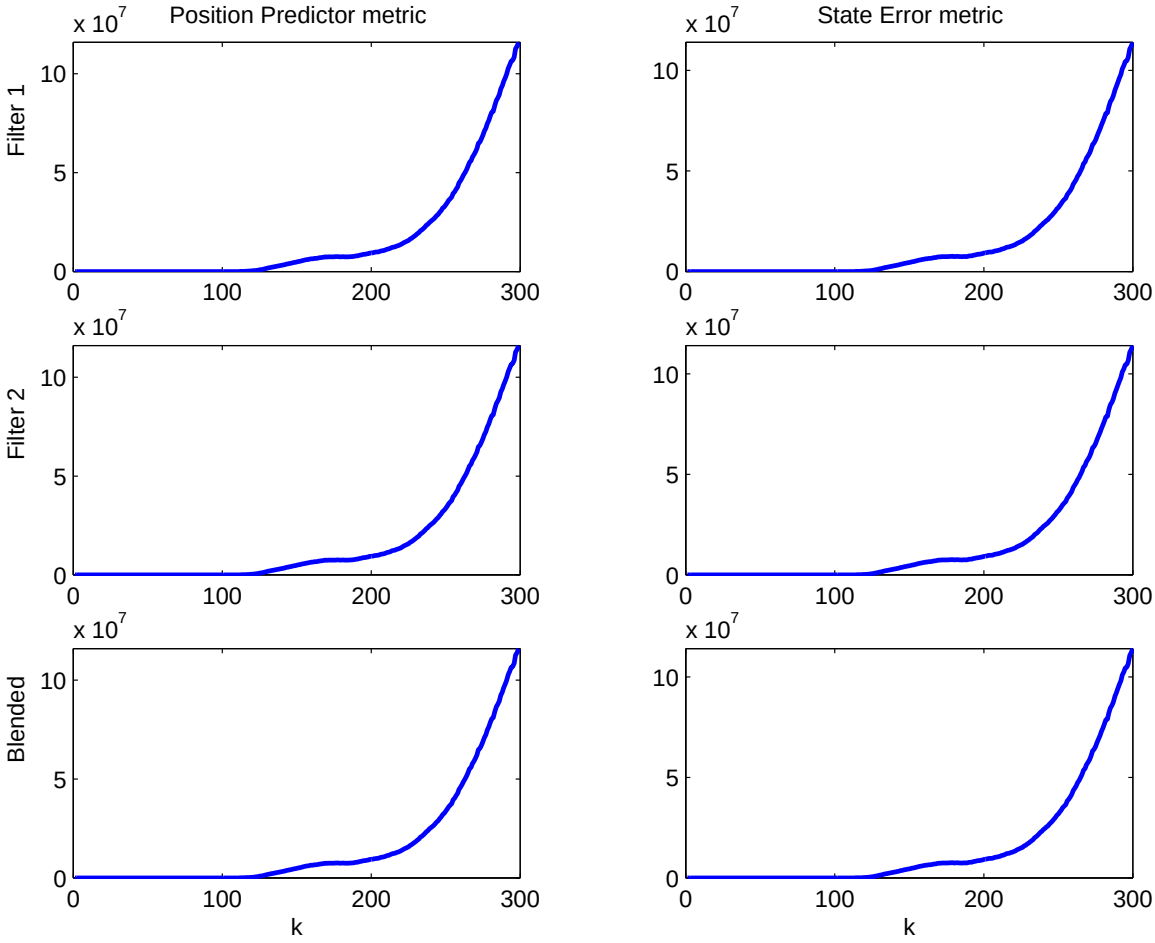

Figure C.4: Track loss results, low-clutter trials. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

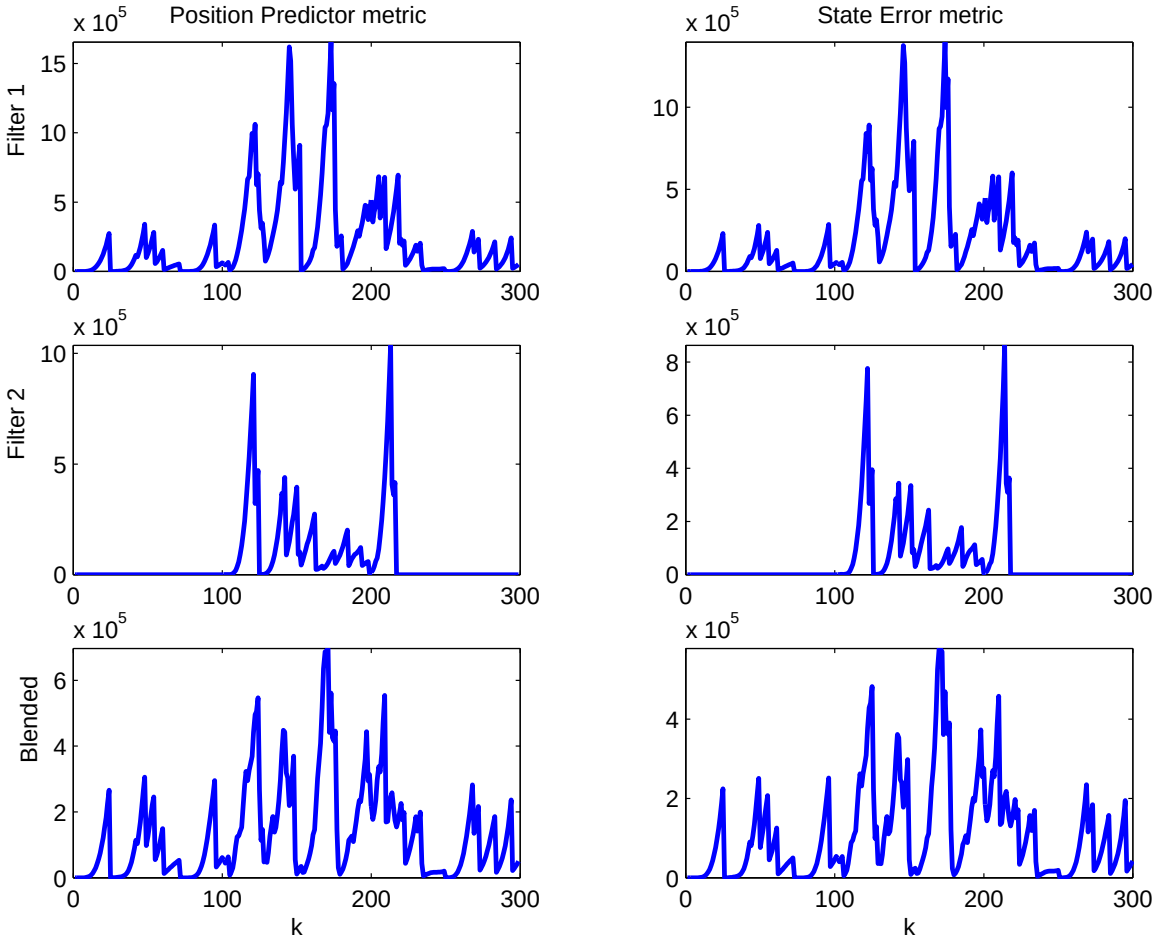

Figure C.5: Track loss results, medium-clutter trials. (Suite 4 vs. Scenario 4: MMAE)

especially in filter 2. The IMM exhibited extremely good performance here, and its track loss quantities are less than one order-of-magnitude larger than they were in the clutter-free case. Maneuver phases are very evident. Especially take note that it is better than the same configuration operating in the low-clutter environment, and note that MMAE performance in the same medium-clutter environment was slightly worse than it was in the low-clutter environment. As discussed in Chapter 5, this IMM case is probably representative of a particular good measurement gating environment. It is recommended that the reader understand the implications of this configuration by reading Section 5.2.

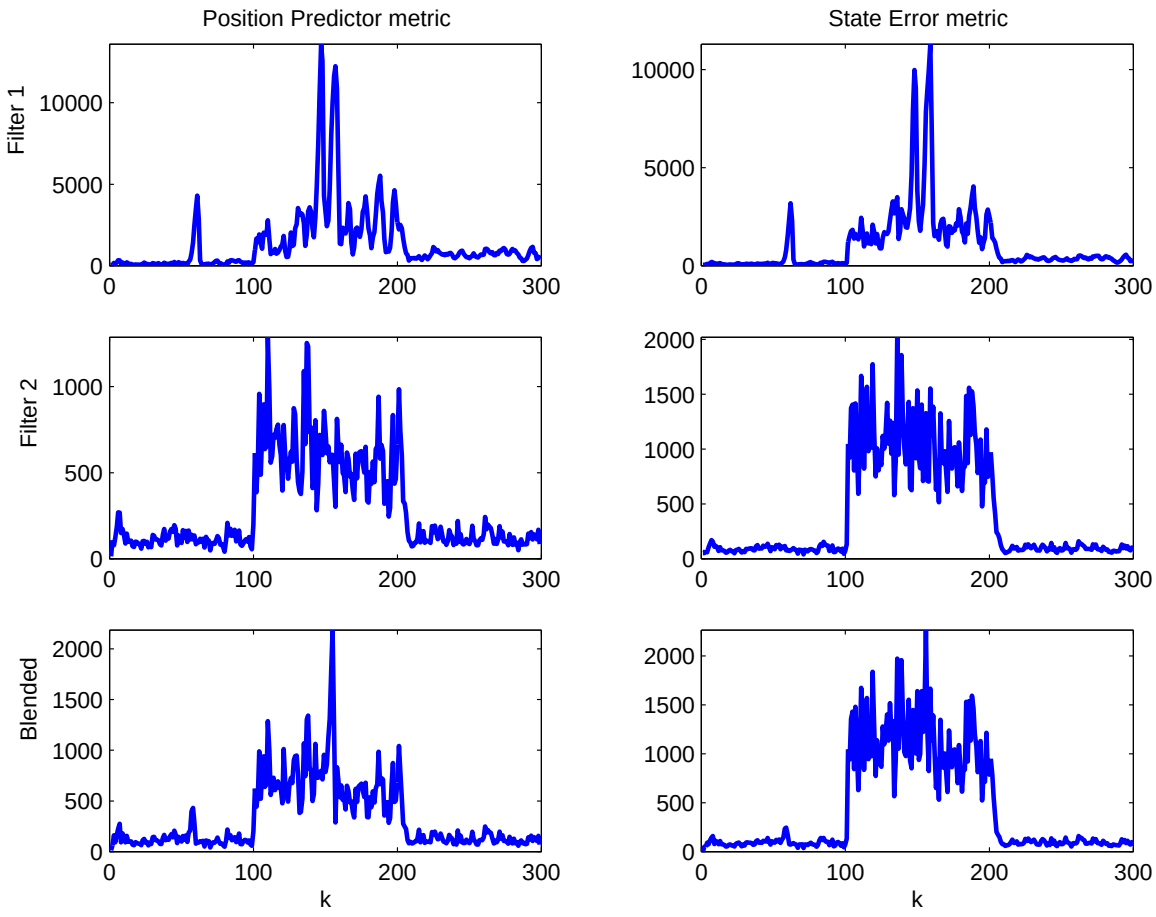

Figure C.6: Track loss results, medium-clutter trials. (Suite 4 vs. Scenario 4: IMM (Non-Trans. Favor. Markov))

C.3 Suite 5 versus Scenario 5

Scenario 5 is a +3-g TPV maneuver truth model scenario. The clutter-free trial results for the MMAE and IMM appear in Figs. C.7 and C.8, and performance is not surprising. The track loss quantity magnitudes are larger during the TPV maneuvers than they are during the TPV maneuvers for the case of Suite 6 matched to Scenario 6 (which also has +3-g TPV maneuvers; to be discussed in Section C.4). This is likely an effect of differences in filter-computed covariances more than it is an effect of actual state estimate error.

The low-clutter MMAE (Fig. C.9) and the low-clutter IMM (Fig. C.10) exhibit rather unusual performance. TPV maneuvers are virtually impossible to resolve in either of the outputs. MMAE filters 1 *and* 2 apparently track very well until the last 100 seconds of simulation, at which point all truth model TPV maneuvers cease. Filters 1 and 2 represent mirrored (±3-g) TPV filters, and should normally have very different state estimate error performance at the same sample period. After 275 seconds, the constant-velocity filter 3 track estimate improves substantially, although this represents a significant delay relative to the cessation of TPV maneuvers (which end at 200 seconds). IMM exhibits similar performance, but the intermixing of states makes specific behavior somewhat different. Notice differences in plot scaling make comparisons between IMM and MMAE difficult. The IMM blended estimate *does* appear to regain track faster than MMAE in the last 100 seconds of simulation.

At the medium-clutter case, only MMAE was tested due to time constraints, and the results are shown in Fig. C.11. The track loss checks appear vaguely similar to the MMAE low-clutter case, although the magnitudes prior to 200 seconds are substantially higher, indicating worse track performance. Again, after the cessation of TPV maneuvers at 200 seconds, the MMAE appears to regain track, this time much more quickly than it did in the low-clutter case.

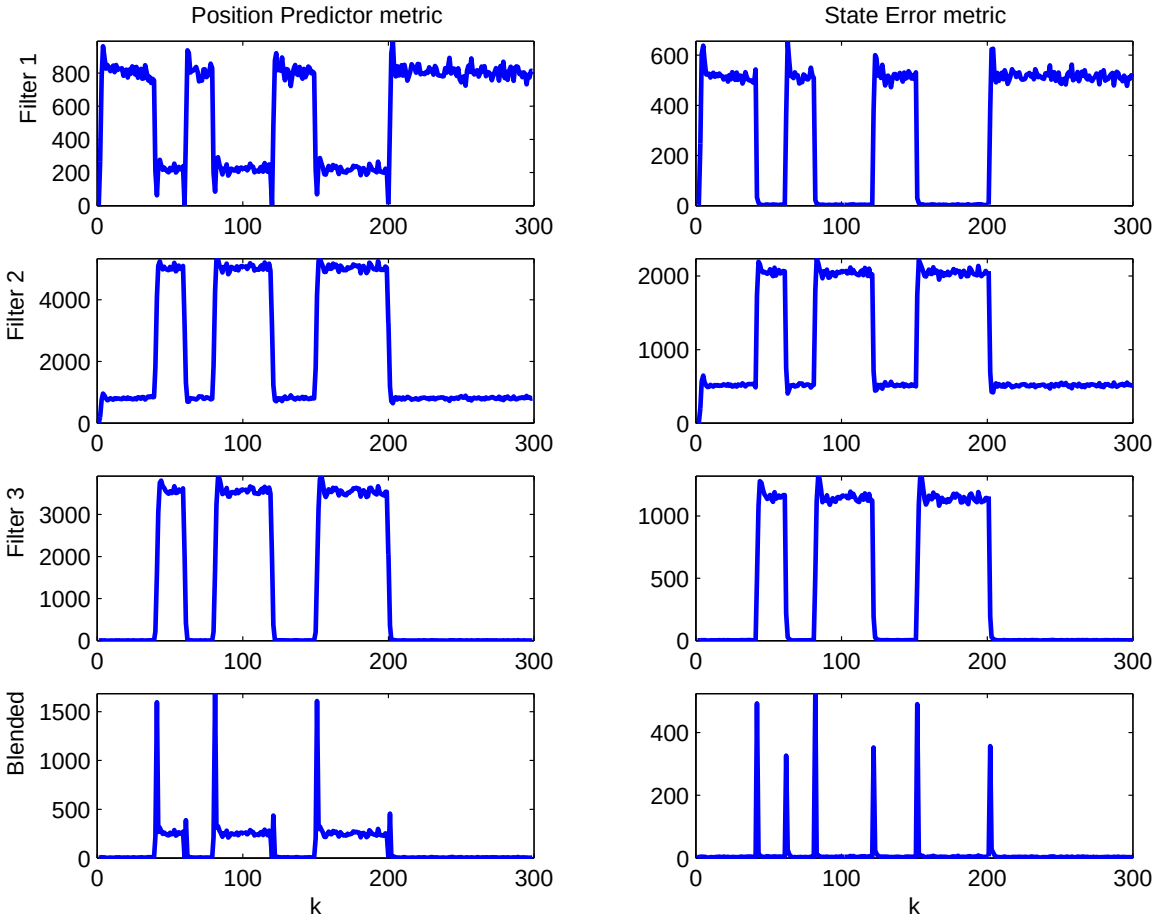

Figure C.7: Track loss results, clutter-free trials. (Suite 5 vs. Scenario 5: MMAE)

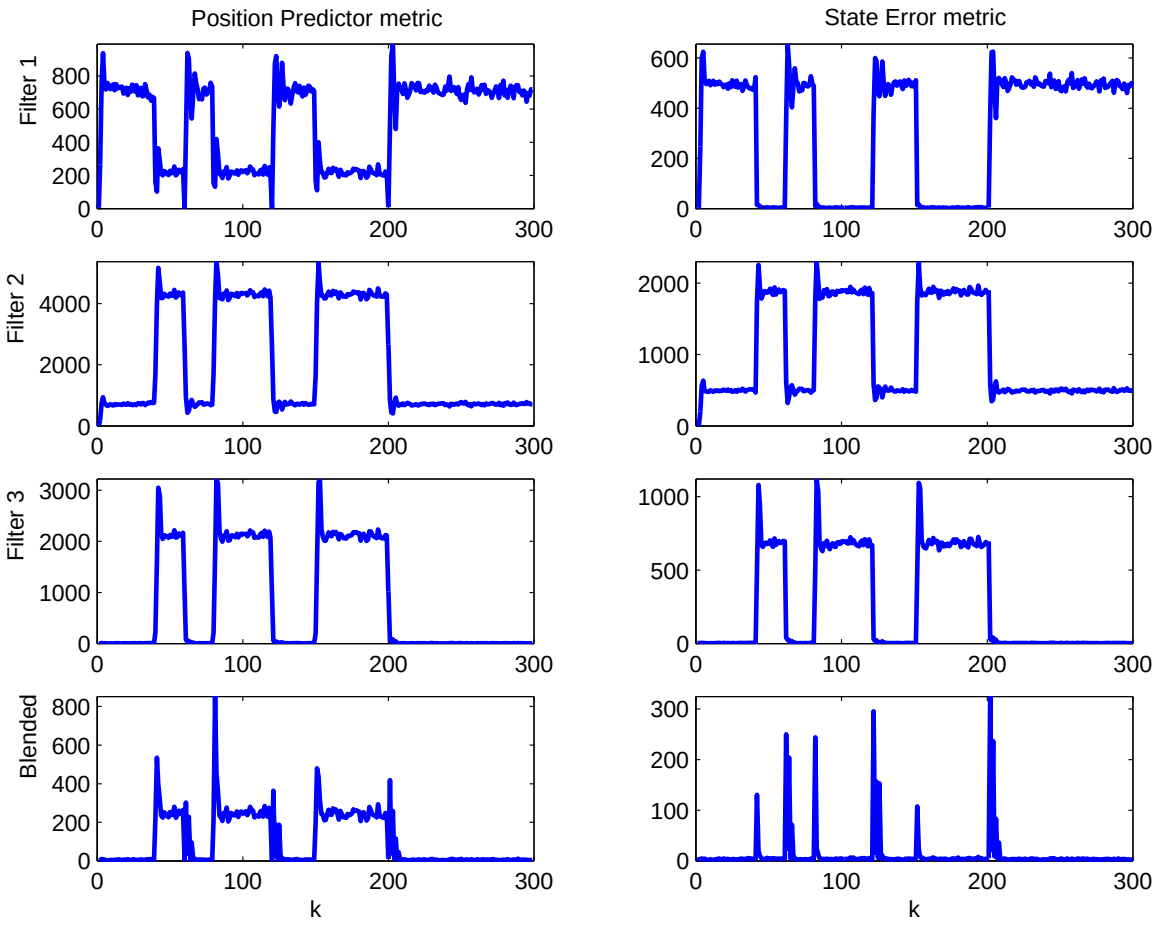

Figure C.8: Track loss results, clutter-free trials. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

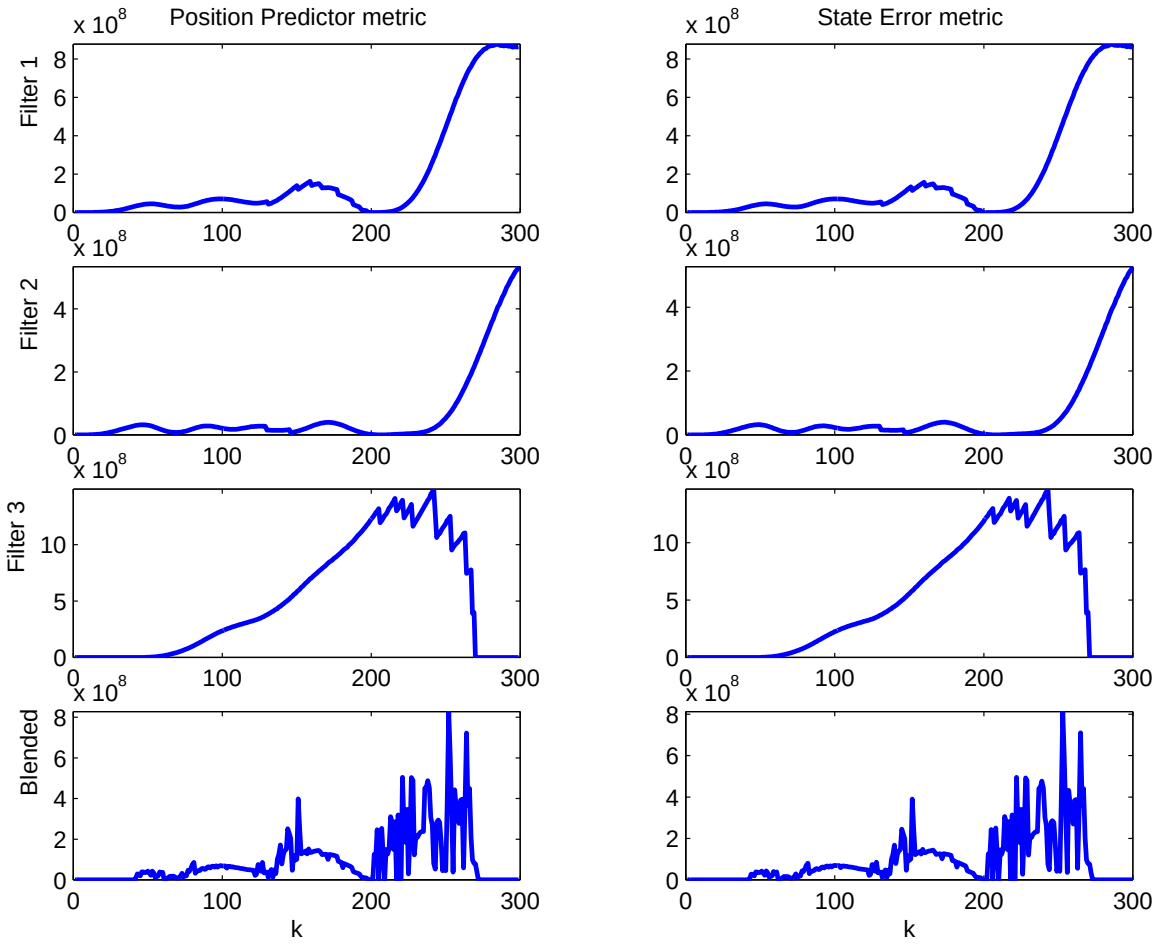

Figure C.9: Track loss results, low-clutter trials. (Suite 5 vs. Scenario 5: MMAE)

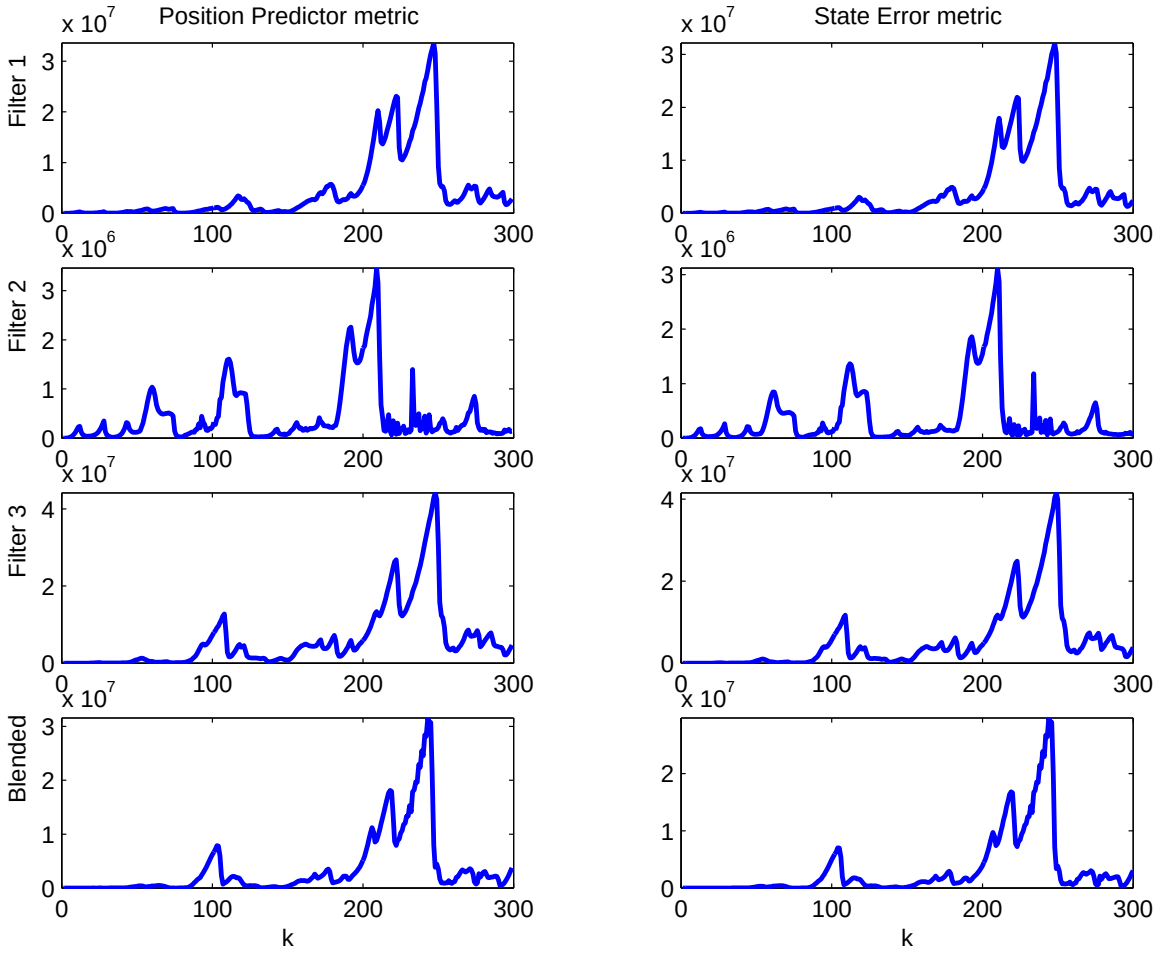

Figure C.10: Track loss results, low-clutter trials. (Suite 5 vs. Scenario 5: IMM (Non-Trans. Favor. Markov))

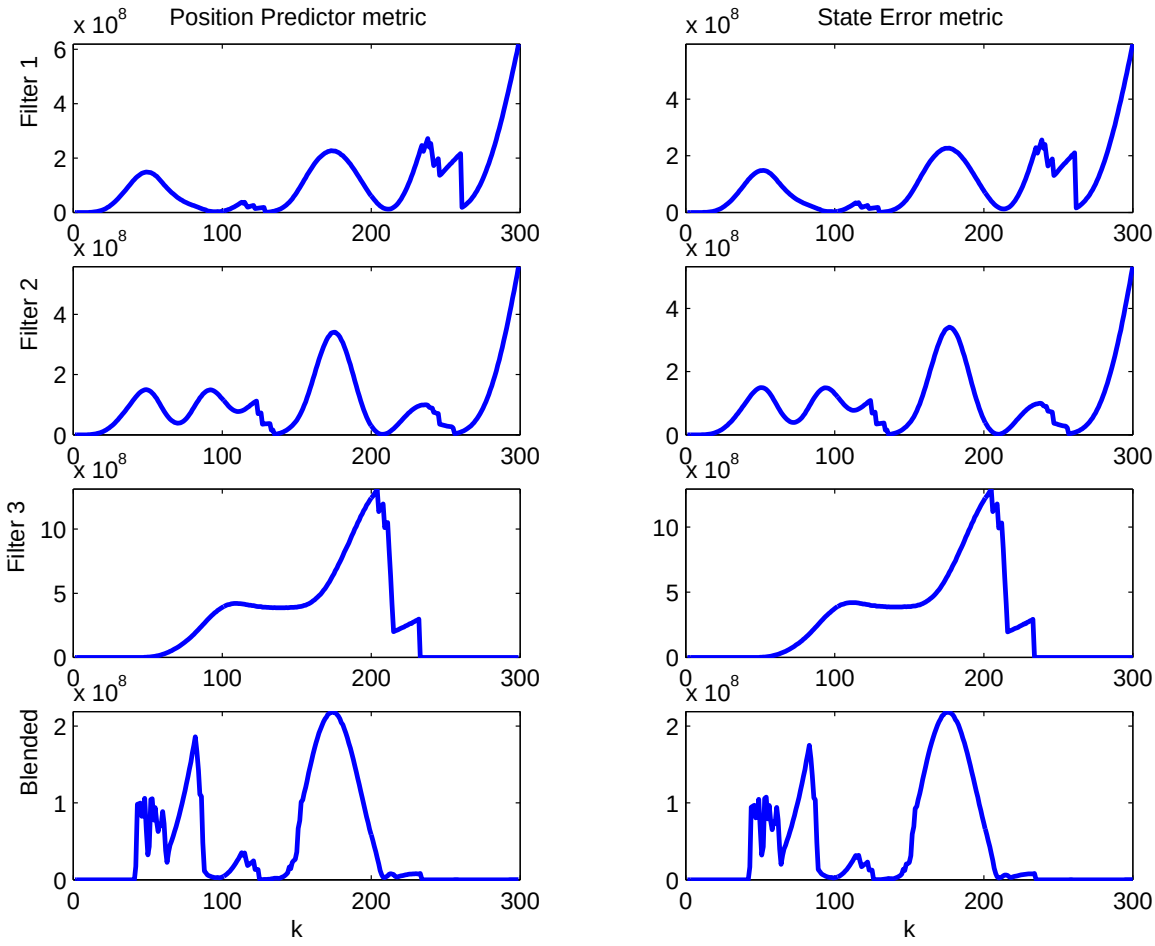

Figure C.11: Track loss results, medium-clutter trials. (Suite 5 vs. Scenario 5: MMAE)

C.4 Suite 6 versus Scenario 6

The clutter-free results for MMAE and IMM appear in Figs. C.12 and C.13, respectively. As expected, the maneuver phases are very evident, but both trackers performed on-par or better than they did in the Scenario 4 trials. During aggressive maneuver phases, the blended estimate magnitudes have a mean somewhere around 50 to 75, with transient spikes going as high as about 500. It is obvious which filters match the truth model during each phase of simulation, and the track loss quantities rapidly adjust as the target maneuver situation changes.

Results for the low-clutter cases appear in Figs. C.14 and C.15. The TPV maneuvers obviously make the tracker's job difficult, evidenced by the large magnitude spikes, primarily in MMAE filter 3 and the blended estimate. Nonetheless, both filter 3 and the blended estimate recovered from these periods of track loss, as evidenced by the very small magnitudes after ($k = 200$) in both the Position Predictor and State Error metrics. Filter 2 diverged significantly prior to ($k = 150$), and at that point, the filter was either reset by the MMAE itself or its internal Gaussian mixture snapped to a new track solution. Filters 1 and 2 both suffered divergence by the end of the simulation, although this was not totally unexpected since no TPV maneuvers occur after ($k = 200$).

IMM appeared to perform worse (notice the order-of-magnitude difference in plot scalings between the MMAE's blended estimate in Fig. C.14 and the IMM's blended estimate in C.15), and particularly evident is the similarity between all of the filter estimates and the blended estimate. This is an effect of the IMM's intermixing of state estimates at every sample period – if none of the filters are performing particularly well on their own, each filter begins to perform about the same because the mixing probabilities distribute all filter estimates approximately evenly (see Fig. 4.101). Notice that IMM does not suffer from this problem in the clutter-free environment because the modal probabilities (and therefore mixing probabilities) are much better-defined (see Fig. 4.37).

Results for the medium-clutter cases appear in Figs. C.16 and C.17. The MMAE appears to exhibit no appreciable change in performance over the corresponding low-clutter case. The IMM performs noticeably better than it did in the low-clutter case, although each filter and the blended estimate exhibit almost indistinguishable performance. Compared to MMAE, the three maneuver phases, each of which is an identical +3-g TPV maneuver, cause marked differences in performance at each phase. The track loss quantities are significantly larger during the first two maneuvers than in the third maneuver. This behavior is not evident in the low-clutter case, so it is likely a result of the particular clutter measurement environment encountered by the tracker in these trials. Furthermore, although IMM appears to regain track after approximately sample 225, considerable track divergence appears towards the end of simulation.

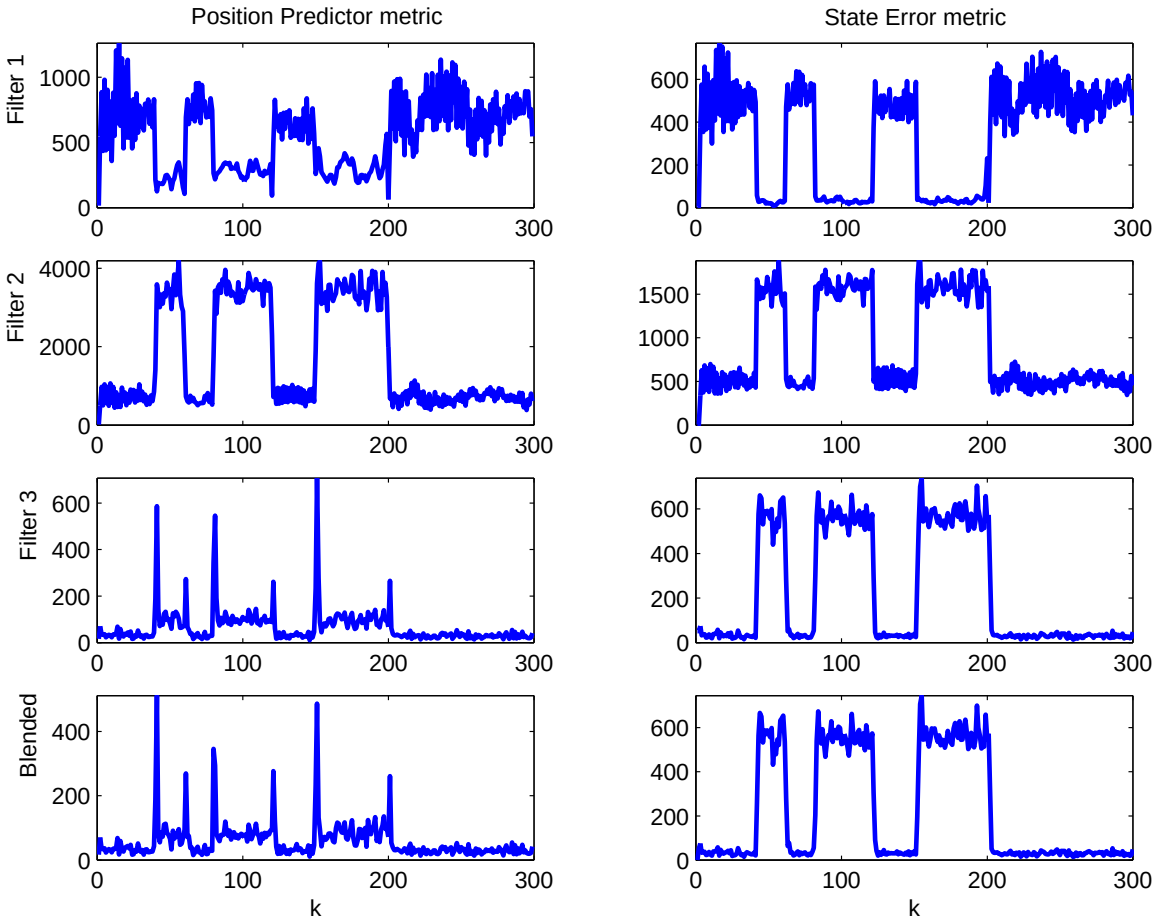

Figure C.12: Track loss results, clutter-free trials. (Suite 6 vs. Scenario 6: MMAE)

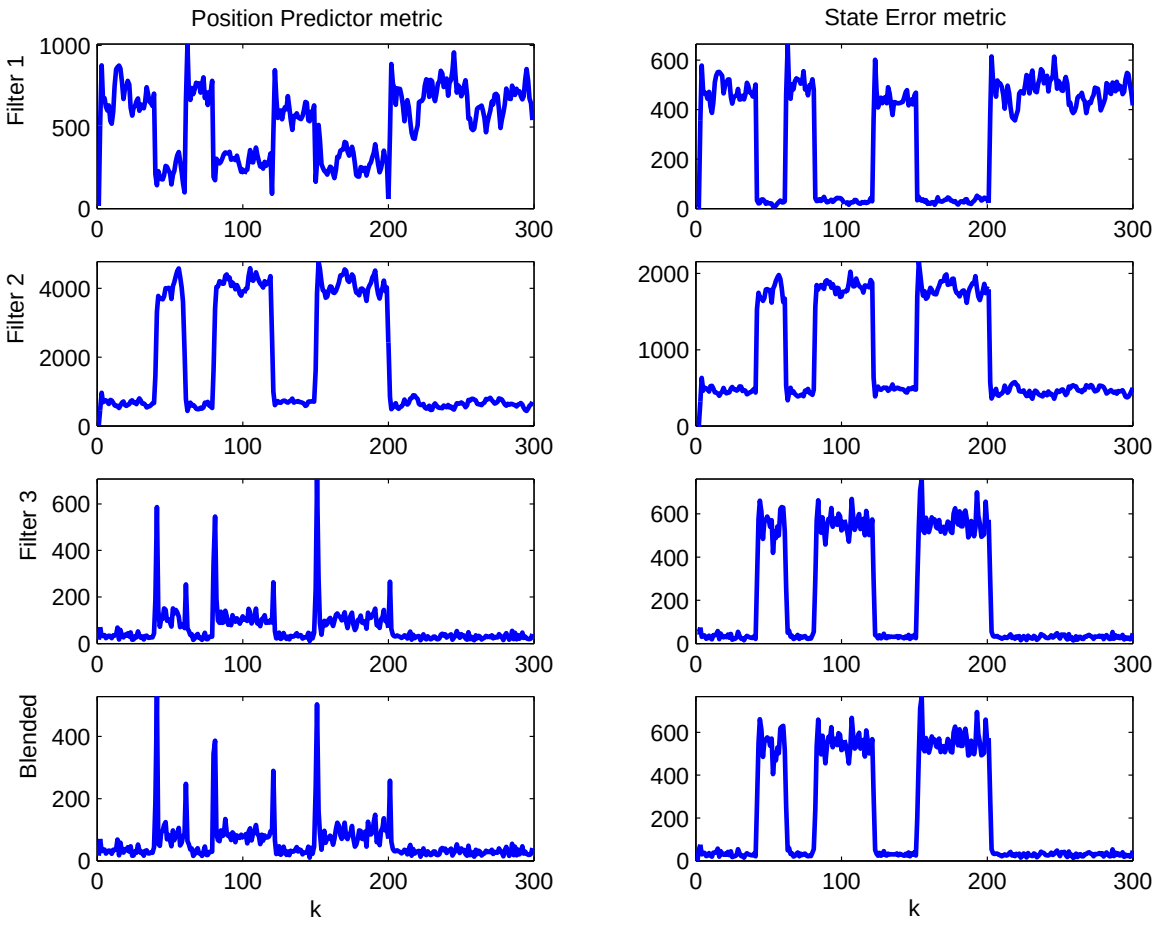

Figure C.13: Track loss results, clutter-free trials. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

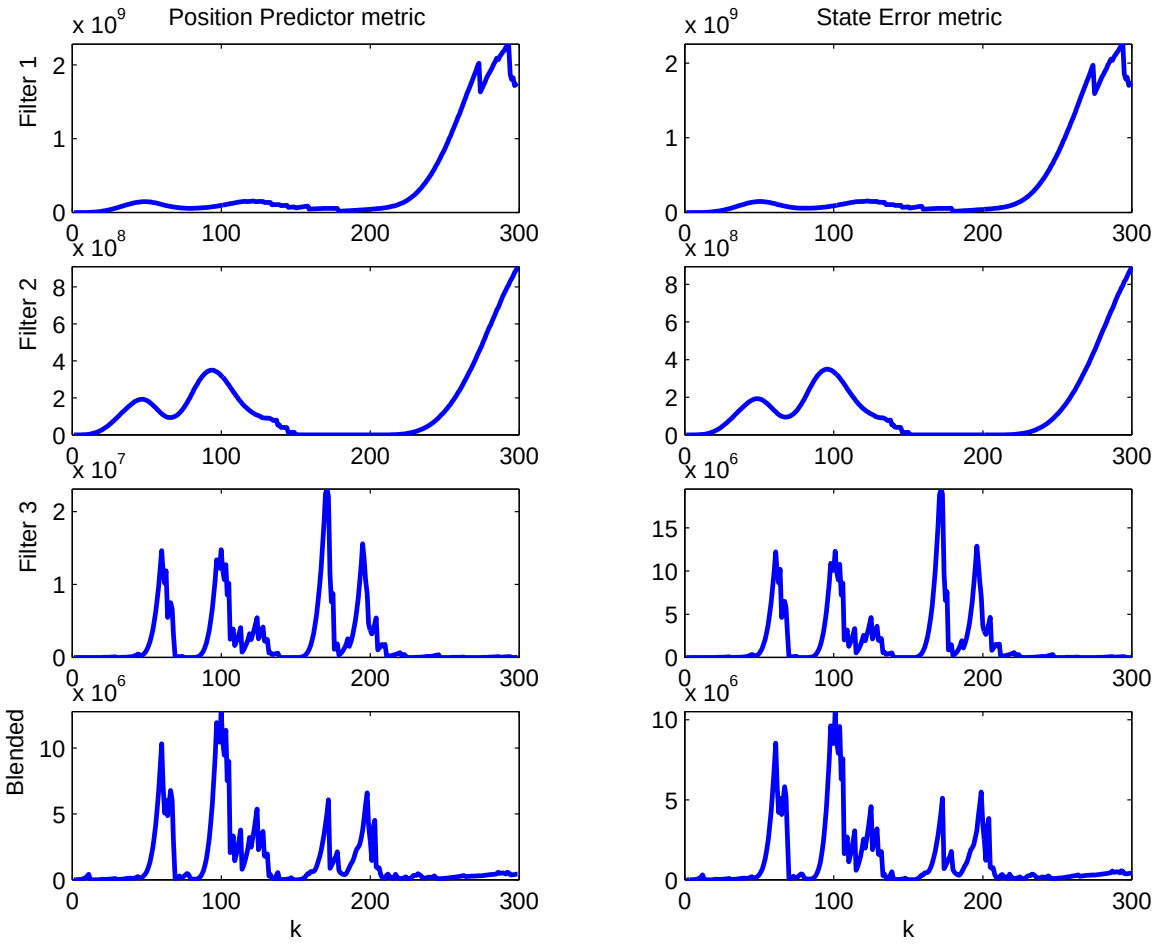

Figure C.14: Track loss results, low-clutter trials. (Suite 6 vs. Scenario 6: MMAE)

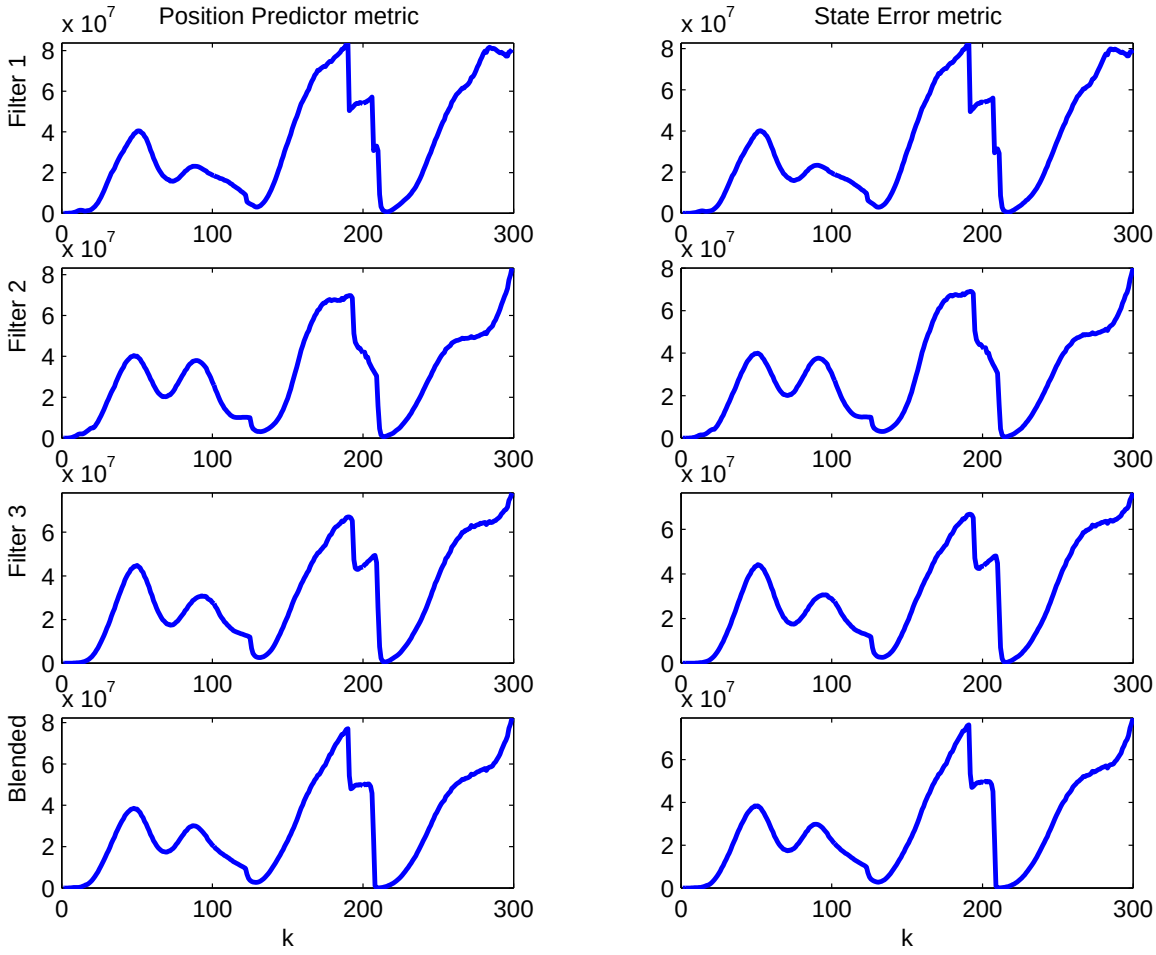

Figure C.15: Track loss results, low-clutter trials. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

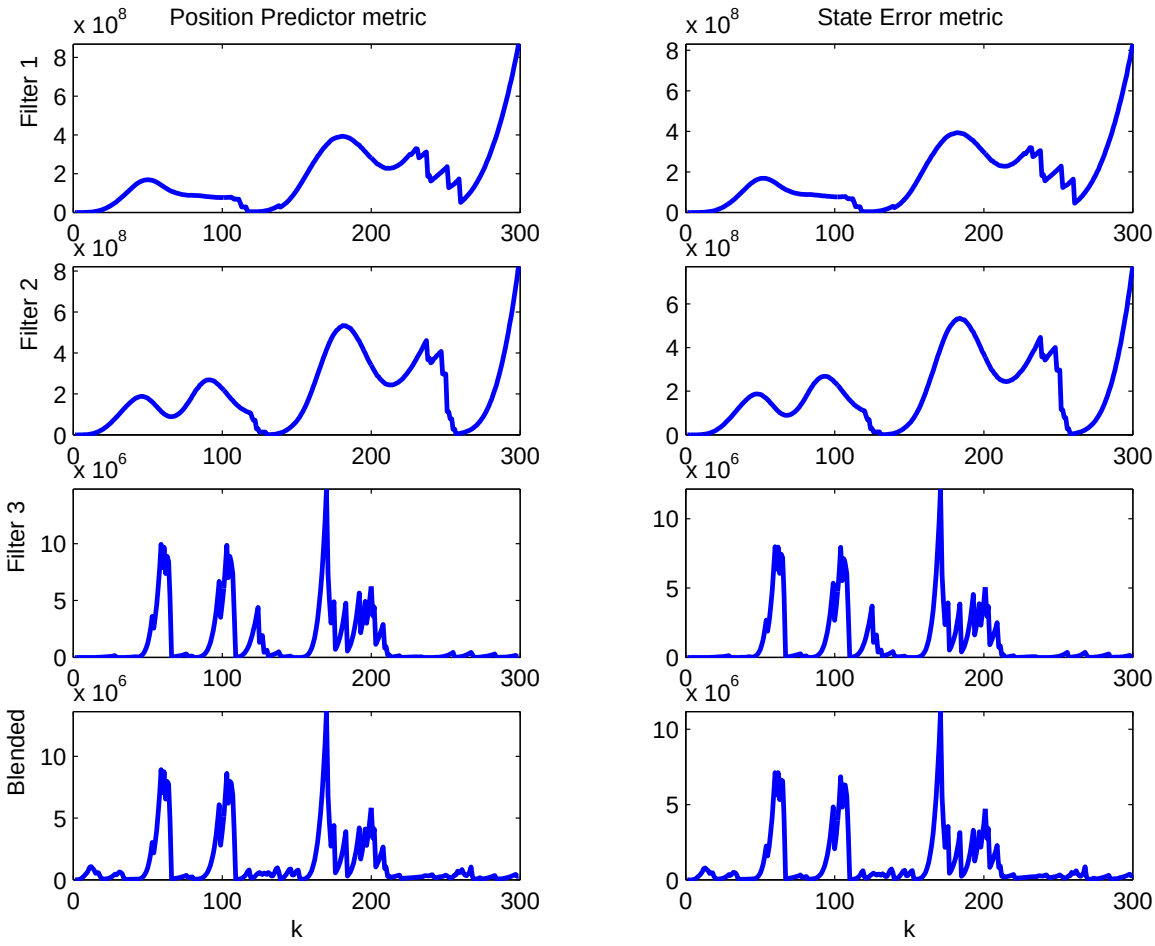

Figure C.16: Track loss results, medium-clutter trials. (Suite 6 vs. Scenario 6: MMAE)

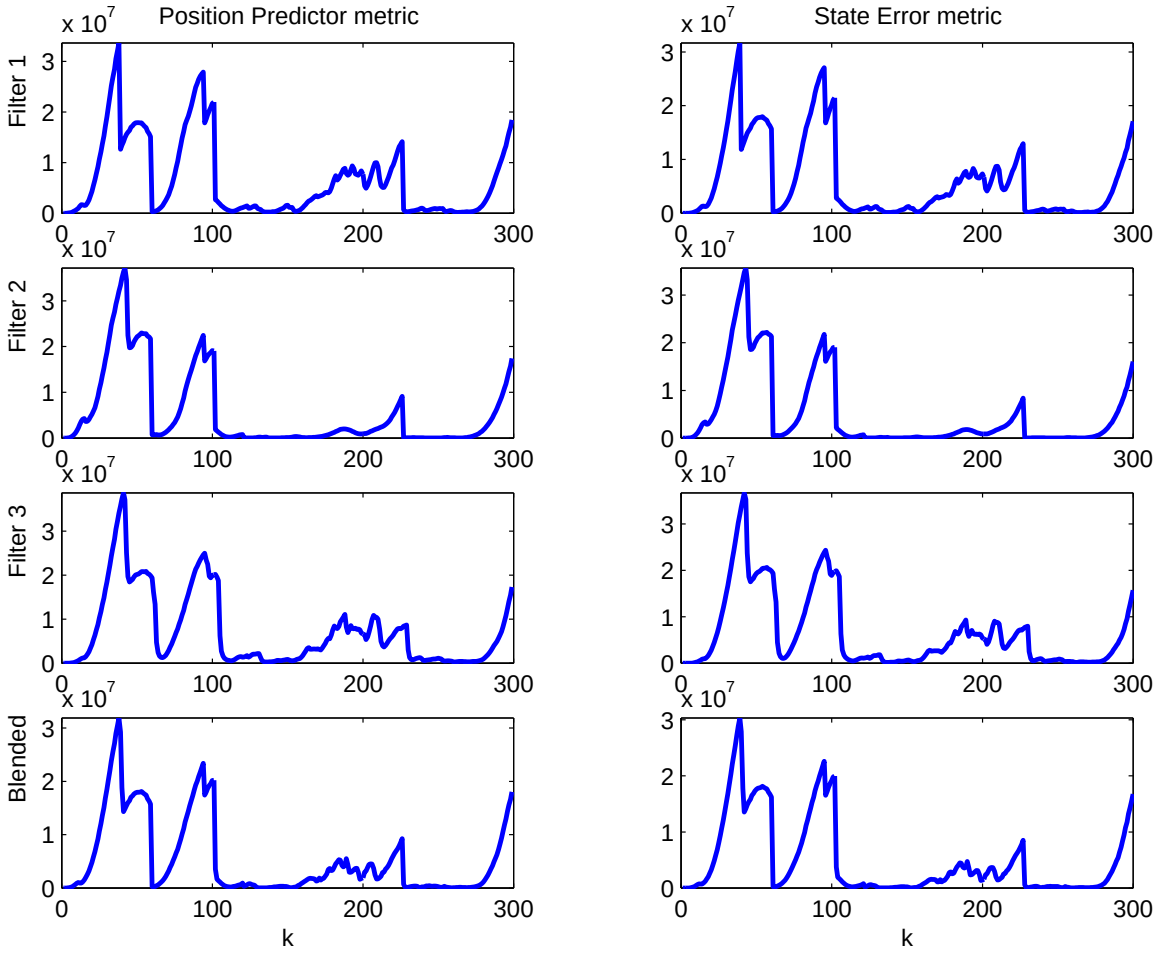

Figure C.17: Track loss results, medium-clutter trials. (Suite 6 vs. Scenario 6: IMM (Non-Trans. Favor. Markov))

C.5 Summary of Track Loss Analysis

The position prediction quadratic and the conventional state estimate error quadratic appear to produce almost equivalent results even though the magnitudes will be different for the sample set of data evaluated at equivalent sample times. Neither of them is particularly difficult to compute off-line once the data has been collected.

Both of these measures to a good job of illuminating changes in truth model dynamics (e.g., during an aggressive maneuver), whereas state estimation error alone may not reveal such maneuvers. Like probability flow plots, these measures can show rapid changes in tracker behavior. The effect of filter-computed covariances in these computations is significant, and cases in which the filter-computed covariances do not adequately represent the actual covariances (e.g., a poorly-tuned filter) may exhibit rather skewed results. Regardless, these quadratic quantities do provide some additional insight into tracker performance. Unlike the mostly qualitative analysis in Chapter 4, these quantities could be used as the basis of design criteria for a real system.

Future research that incorporates these measures should also draw some sort of connection between the track loss measures and a practical "loss of track" declaration. That is to say, some decision should be made regarding what Position Predictor or State Error metric magnitudes constitute an effective loss of track. It is possible that some connection could also be made between the magnitude of these measures and the ability of a tracker to *regain* lock after more measurements arrive. This might allow some quantitative comparison of MHT deferred decision-making ability.

Finally, analysis of single Monte Carlo trials using these measures might yield different insights into tracker behavior than those yielded by the simulation summary plots used in this research (which plotted the mean of the track loss measures over all Monte Carlo trials at a particular sample period). Plotting the mean of these quantities gives an overall picture of the algorithm's performance, but it folds divergent runs and non-divergent runs into the same track loss measure. For a single-run analysis to be meaningful, one would probably want to characterize the simulation environment at the time of track loss (e.g., filter gate sizes, number of measurements within the gates, relative position of the elemental filter estimates inside the measurement space, etc.) so that patterns of algorithm behavior could be identified. Using only the mean plots, it is difficult to isolate which situations create the most problems for the MHT.

Bibliography

1. Alspach, D.L. "A Gaussian Sum Approach to the Multitarget Identification–Tracking Problem". *Automatica*, 11(3):285–296, May 1975.

2. Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp. "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking". *IEEE Transactions on Signal Processing*, 50:174–188, February 2002.

3. Bar-Shalom, Yaakov and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Norwood, MA, 1993.

4. Bar-Shalom, Yaakov and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT, 1995.

5. Blackman, Samuel S. and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.

6. Blom, Henk A.P. and Yaakov Bar-Shalom. "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients". *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.

7. Blom, Henk A.P. and Edwin A. Bloem. "Joint Probabilistic Data Association Avoiding Track Coalescence". *IEE Colloquium on Algorithms for Target Tracking*, 1/1–1/3, May 1995.

8. Chen, R. and J. S. Liu. "Mixture Kalman Filters". *Journal of the Royal Statistical Society*, B 62(3):493–508, 2000.

9. Dempster, R.J., S.S. Blackman, and T.S. Nichols. "Combining IMM Filtering and MHT Data Association for Multitarget Tracking". *Proceedings of the 29th Southeastern Symposium on System Theory*, 123–127. IEEE Press, Cookeville, TN, March 1997.

10. Eide, Peter and Peter S. Maybeck. "An MMAE Failure Detection System for the F-16". *IEEE Transactions on Aerospace and Electronic Systems*, 32, No. 3:1125–1136, July 1996.

11. Gustafson, John A. and Peter S. Maybeck. "Flexible Spacestructure Control Via Moving-Bank Multiple Model Algorithms". *IEEE Transactions on Aerospace and Electronic Systems*, 30, No. 3:750–757, July 1994.

12. Izenman, A. J. "Recent Developments in Nonparametric Density Estimation". *Journal of the American Statistical Association*, 86:205–224, March 1991.

13. Kastella, Keith. "A Maximum Likelihood Estimator for Report-to-Track Association". *SPIE Signal and Data Processing of Small Targets*, 1954:386–393, October 1993.

14. Kirubarajan, T., Y. Bar-Shalom, W.D. Blair, and G.A. Watson. "IMMPDAF for Radar Management and Tracking Benchmark with ECM". *IEEE Transactions on Aerospace and Electronic Systems*, 34, No. 4:1115–1134, October 1998.

15. Korn, J. and L. Beean. *Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation*. Technical report, Alphatech, Inc., Burlington, MA, June 1983.

16. Kyger, David W. and Peter S. Maybeck. "Reducing Lag in Virtual Displays Using Multiple Model Adaptive Estimation". *IEEE Transactions on Aerospace and Electronic Systems*, 34, No. 4:1237–1248, October 1998.

17. Maybeck, Peter S. *Stochastic Models, Estimation, and Control*, volume Volume 2. Navtech, Arlington, VA, 1994.

18. Maybeck, Peter S. *Stochastic Models, Estimation, and Control*, volume Volume 1. Navtech, Arlington, VA, 1994.

19. Maybeck, Peter S. and Steven K. Rogers. "Adaptive Tracking of Multiple Hot-Spot Target IR Images". *IEEE Transactions on Automatic Control*, 28, No. 10:937–943, October 1983.

20. Menke, Timothy E. and Peter S. Maybeck. "Sensor/Actuator Failure Detection in the Vista F-16 by Multiple Model Adaptive Estimation". *IEEE Transactions on Aerospace and Electronic Systems*, 31, No. 4:1218–1229, October 1995.

21. Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, second edition, 1992.

22. Reid, Donald B. "An Algorithm for Tracking Multiple Targets". *IEEE Transactions on Automatic Control*, AC-24(6):843–854, December 1979.

23. Salmond, David J. *Mixture Reduction Algorithms for Uncertain Tracking*. Technical Report 88004, Royal Aerospace Establishment, Farnborough, UK, January 1988. DTIC Number ADA197641.

24. Salmond, David J. "Mixture Reduction Algorithms for Target Tracking". *IEE Colloquium on State Estimation in Aerospace and Tracking Applications*, 7/1–7/4. IEE Publishing, London, UK, December 1989.

25. Salmond, David J. *Tracking in Uncertain Environments*. Technical Memorandum AW 121, Royal Aerospace Establishment, Farnborough, UK, September 1989. DTIC Number ADA215866. Taken from a D Phil thesis of the University of Sussex.

26. Salmond, David J. "Mixture Reduction Algorithms for Target Tracking in Clutter". *SPIE Signal and Data Processing of Small Targets*, 1305:434–445, April 1990.

27. Scott, D. W. "Remarks on Fitting and Interpreting Mixture Models". *Proceedings of the 31st Symposium on the Interface of Computing and Statistics*, 104–109. Interface Foundation of North America, Fairfax, VA, 1999.

28. Singer, R.A., R.G. Sea, and K.B. Housewright. "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multi-target Environments". *IEEE Transactions on Information Theory*, IT-20(4):423–832, July 1974.

29. Strang, G. *Linear Algebra and its Applications*. Harcourt College Publishers, Orlando, FL, third edition, 1988.

30. Williams, Jason L. *Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter*. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2003. AFIT/GE/ENG/03-19.

31. Williams, Jason L. and Peter S. Maybeck. "Cost-Function-Based Hypothesis Control Techniques for Mulitple Hypothesis Tracking". *Proceedings of the SPIE Annual International Defense and Security Symposium, Vol. 5428*, 167–179. SPIE Press, Orlando, FL, April 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–03–2005	Master's Thesis	Sept 2003 — Mar 2005

4. TITLE AND SUBTITLE

Multiple Model Adaptive Estimator Target Tracker
For Maneuvering Targets in Clutter

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Brian D. Smith, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management, Bldg. 641
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/05-18

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Stanton H. Musick
AFRL/SNAT
2241 Avionics Circle
WPAFB, OH 45433-7333
DSN: 785-1115 x4292

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT In recent years, the Multiple Hypothesis Tracker has gained acceptance as means of handling targets in a measurement-clutter environment. MHT algorithms rely on Gaussian mixture representations of a target's current state estimate, and the number of components within these mixtures grows exponentially with each successive sensor scan. Previous research into techniques that limit the growth of Gaussian mixture components proved that the Integral Square Error cost-function-based algorithm performs well in this role. Also, multiple-model adaptive algorithms have been shown to handle poorly-known target dynamics or targets that exhibit a large range of maneuverability over time with excellent results. This research integrates the ISE mixture reduction algorithm into Multiple-Model Adaptive Estimator (MMAE) and Interacting Mixed Model (IMM) tracking algorithms. The algorithms were validated to perform well at a variety of measurement clutter densities by using a Monte Carlo simulation environment based on the C++ language. Compared to single-dynamics-model MHT trackers running against a maneuvering target, the Williams-filter-based multiple-model algorithms exhibited superior tracking performance.

15. SUBJECT TERMS

Multiple Hypothesis Tracker, multiple-model estimation, multiple model adaptive estimator, interacting mixed model, Kalman filter, Williams filter

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Peter S. Maybeck
U	U	U	UU	308	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 4581